Annals of Pure and Applied Logic 161 (2010) 1254–1269

Contents lists available at ScienceDirect

Annals of Pure and Applied Logic

journal homepage: www.elsevier.com/locate/apal

Games with 1-backtracking

Stefano Berardi [a,*], Thierry Coquand [b], Susumu Hayashi [c]

[a] Department of Computer Science, University of Turin, Turin, Italy
[b] Department of Computer Science, Chalmers University, Goteborg, Sweden
[c] Department of Humanistic Informatics, Kyoto University, Kyoto, Japan

ARTICLE INFO

Article history:
Received 16 March 2007
Received in revised form 25 February 2010
Accepted 6 March 2010
Available online 1 May 2010
Communicated by Guest Editor (ESNL)

MSC:
03B30
91D35
03D35

Keywords:
Classical logic
Game semantics
Backtracking
Learning in the limit
Limit computable
Recursive degree

ABSTRACT

We associate with any game G another game, which is a variant of it, and which we call $\mathtt{bck}(G)$. Winning strategies for $\mathtt{bck}(G)$ have a lower recursive degree than winning strategies for G: if a player has a winning strategy of recursive degree 1 over G, then it has a recursive winning strategy over $\mathtt{bck}(G)$, and vice versa. Through $\mathtt{bck}(G)$ we can express in algorithmic form, as a recursive winning strategy, many (but not all) common proofs of non-constructive Mathematics, namely exactly the theorems of the sub-classical logic Limit Computable Mathematics (Hayashi (2006) [6], Hayashi and Nakata (2001) [7]).

1. Introduction

We consider a class of games with possibly infinite plays. We associate with any game G in this class another game in the same class, a variant of G which we call $\mathtt{bck}(G)$. We plan to use games of the form $\mathtt{bck}(G)$ as a semantics for a sub-classical logic, Limit Computable Mathematics, or LCM for short (see [6] and Appendix).

In the terminology of [4,5], $\mathtt{bck}(G)$ consists of all plays with backtracking over G, whose Interaction Sequence (f, V) satisfies $\forall x > 0.\exists n.f(x) = f^n(x-1)$. In the terminology of Hyland–Ong [9], $\mathtt{bck}(G)$ consists of all plays in which both players use innocent strategies always pointing within the current thread of the play. The interest of $\mathtt{bck}(G)$, however, lies in the fact that it also has a simple and direct definition, as we will explain. We first describe $\mathtt{bck}(G)$, then we sketch the use we have in mind for it.

In the case G is the game of Chess, we can informally describe $\mathtt{bck}(G)$ as follows. $\mathtt{bck}(G)$ is a variant of Chess we could call "Chess with backtracking", in which both players can "learn" better moves by trial-and-error. In Chess with backtracking, each player can decide that he has made a mistake at some previous move, and retract it. All moves in between are then forgotten. A motivation for this game could be teaching Chess to beginners. If you suffer checkmate in Chess with backtracking, you do not necessarily lose: you can come back to some previous position of the chessboard, and select a move from the position you consider to be better. However, each player is allowed to improve only finitely many times the move

* Corresponding author.
E-mail addresses: stefano@di.unito.it (S. Berardi), coquand@cs.chalmers.se (T. Coquand), susumu@shayashi.jp (S. Hayashi).

0168-0072/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.apal.2010.03.002

from any given position of the chessboard. The first player changing a move infinitely many times loses. We give a name to this "retracting": we call it "1-backtracking". Both players can perform "1-backtracking". The aim of 1-backtracking is to learn better moves. We call $\mathtt{bck}(G)$ the "1-backtracking" version of G.

In $\mathtt{bck}(G)$, as in Chess with backtracking, a player can come back to the same position x of G only finitely many times, otherwise he loses. The idea is that a player can change several times his move from x in order to make it progressively better. A player's first choice may be thoughtless, but his next choice depends on his previous experience in the play and will be better. If the player makes some mistakes, again he can exploit his experience to select some even better move, and so on. However, any player has only a finite time to improve his move, and eventually he must stop changing his mind. He must provide, for any position x, a last and definitive choice for his move from x. We require that there is a last move from the initial position of G, a last move from this last move, and so forth. In this way, every play β in $\mathtt{bck}(G)$ is associated with some play $\beta^{(1)}$ in G, made only of "last moves" (Definition 7.4). As usual, a player who does not move, or who does an irregular move loses. Using $\beta^{(1)}$, we can define the winner of a possibly infinite play β in $\mathtt{bck}(G)$: it is the winner of the associated play $\beta^{(1)}$ in G.

In spite of the great difference between G and $\mathtt{bck}(G)$, if a player has a winning strategy for G, then he has a winning strategy for the $\mathtt{bck}(G)$, and viceversa. Besides, winning strategies for $\mathtt{bck}(G)$ have a lower recursive degree than winning strategy for G. We now explain what this latter statement means.

Assume that $\mathcal{O}$ is any recursive degree and $\mathcal{O}'$ is the jump of $\mathcal{O}$. We will include the definitions of recursive degree and jump in Section 6: for the moment, the reader may assume that $\mathcal{O} = 0 =$ the set of recursive maps and $\mathcal{O}' = 1 =$ the set of the maps recursive in the Halting Problem. Fix any game G which can be described in $\mathcal{O}$. Assume either G is a Tarski game, or players alternate in G.[1] Then we will prove:

Theorem 1. *If a player p has some winning strategy for G, of recursive degree $\mathcal{O}'$, then p has some winning strategy for* $\mathtt{bck}(G)$, *of recursive degree $\mathcal{O}$.*

Theorem 2. *Conversely, if a player p has some winning strategy for* $\mathtt{bck}(G)$, *of recursive degree $\mathcal{O}$, then p has some winning strategy for G, of recursive degree $\mathcal{O}'$.*

The application we have in mind for $\mathtt{bck}(G)$ are Tarski games. Fix any (closed) formula A of Peano Arithmetic in the connectives $\vee$, $\wedge$, $\exists$, $\forall$. Let G be the Tarski game for A (see Section 2 for details). It is known that we have a winning strategy for G if and only if A is true, and we have a recursive winning strategy for A if and only if there is a constructive proof of A. For instance, if A is $\mathtt{EM}_1$ (Excluded Middle for semi-decidable statements, see Section 2), then there is a winning strategy for G, but no recursive winning strategy, because $\mathtt{EM}_1$ is true, but it has no constructive proof. However, significantly, in this case we have a *recursive* winning strategy for $\mathtt{bck}(G)$ (see Section 3). This pattern is exhibited in many well-known combinatorial and algebraic theorems (some examples are given in Section 4). In $\mathtt{bck}(G)$, we do not have to provide a winning move at once, but we can start by choosing some move, and later we can change it, if we lose the game G, using the experience we gathered from the (temporary) defeat. Recursive winning strategies for $\mathtt{bck}(G)$ may be interpreted as algorithms "learning by trial-and-error" winning moves in G.

The idea is therefore to use $\mathtt{bck}(G)$ in order to associate many common non-constructive theorems with some algorithmic content, expressed by the winning strategy for $\mathtt{bck}(G)$. Theorems 1 and 2 above have an interesting corollary: they characterize for which games G we have a recursive winning strategy for $\mathtt{bck}(G)$. If G is a Tarski game we have a recursive winning strategy over $\mathtt{bck}(G)$ if and only if we have a winning strategy of G of recursive degree 1 (i.e., recursive in an oracle for the Halting problem). Even if G and $\mathtt{bck}(G)$ are "equivalent" games, $\mathtt{bck}(G)$ is a version of G for which we have "more concrete" winning strategies. Notice that there are also some recursive games G for which we have winning strategies for G, but no recursive winning strategy for G, nor for $\mathtt{bck}(G)$. This is the case of the game G interpreting $\mathtt{EM}_2$ (Excluded Middle for degree 2 formulas, see Corollary 1.3). This result settles an open problem in [4], p. 5: "1-backtracking" (called "simple backtracking" in [4]) and "backtracking" are *not* equivalent.

Theorems 1 and 2 have another interesting corollary: they characterize true formulas of Limit Computable Mathematics [6] in terms of games. For a brief introduction to $\mathtt{LCM}$ we refer to Appendix. In [6], there is an interpretation for formulas true in $\mathtt{LCM}$. Intuitively, a formula is true in $\mathtt{LCM}$ if we may "learn" that A is true by trial-and-error. "Learnable" formulas are a proper subset of true formulas (for instance, $\mathtt{EM}_2$ is true and not learnable). For this reason, $\mathtt{LCM}$ is also called: "Mathematics based on learning".

Let A be a formula in the connectives $\vee$, $\wedge$, $\exists$, $\forall$, and let G be the Tarski game associated with A. It is known that A is a theorem of $\mathtt{LCM}$ if and only if there is a winning strategy for G of recursive degree 1. If we combine this remark with our result about $\mathtt{bck}(G)$, we conclude that A is a theorem of $\mathtt{LCM}$ if and only if there is a recursive winning strategy for $\mathtt{bck}(G)$. We can interpret this result as follows: we may "learn" A by trial-and-error, if and only if we have a recursive winning strategy with 1-backtracking for the Tarski game associated with A. As far as we know, 1-backtracking games are the first complete characterization of "learnable" formulas in terms of games with *effectively computable* strategies.

This is the plan of the paper. In Section 2 we define the class of games we consider, and we recall the notion of Tarski game. In Section 3 we define $\mathtt{bck}(G)$. In Section 4 we give some examples of games with 1-backtracking. In Section 5 we

[1] In the actual proof of Theorem 2, we replace this hypothesis with a weaker one, Liveness (see Section 5 for a discussion).

discuss Liveness, a generalization of the condition that players alternate in the game. In Section 6 we prove that winning strategies for $\mathtt{bck}(G)$ of degree 0 may be transformed into winning strategies for G of degree 1. This result does not require hypotheses on G. In Section 7, we prove the converse, that winning strategies for G of degree 1 may be transformed into winning strategies for $\mathtt{bck}(G)$ of degree 0. This result requires the Liveness hypothesis on G. In Section 8 we derive the characterization of LCM as corollary. Eventually, in Section 9 we discuss conclusions and future works, and in Appendix we include a brief introduction to LCM.

2. Games and Tarski games

In this section we define the notion of game we consider in this paper, with possibly infinite plays, and the particular case of Tarski games, in which all plays are finite. For a motivation of the definition of game we choose we refer to [10].

We introduce first a notion of tree. We denote the concatenation of two lists x, y by $x@y$. We write $x \leq y$ for "x is a prefix of y", and $x < y$ for $x \leq y$ and $x \neq y$. Trees, in our presentation, are particular sets of lists, as stated in the next definition.

Definition 1 (*Trees and Tree Terminology*)**.**

1. (Trees) A tree T over a set M is some set of lists $\langle *, m_1, \ldots, m_k\rangle$, with $m_1, \ldots, m_k \in M$, including $\langle *\rangle$, and closed under non-empty prefix: if $x \in T$ and $\langle\rangle \neq y \leq x$, then $y \in T$.
2. (Children) $\langle *\rangle$ is the root of T. If $x, x@\langle m\rangle \in T$, we say that x is the *father* of $x@\langle m\rangle$ in T, and that $x@\langle m\rangle$ is a *child* of x in T. Any $x@\langle m'\rangle$ is a *brother* of $x@\langle m\rangle$. If $x \leq y$ we say that x is an *ancestor* of y, and that y is a *descendant* of x. x is a proper ancestor of y and y a proper descendant of x if $x < y$.
3. (Branches) A *leaf* is any $x \in T$ with no children. A *branch* of T is either any $\langle *, m_1, \ldots, m_k\rangle \in T$, or any infinite list $\langle *, m_1, m_2, \ldots, m_k, \ldots\rangle$ whose finite prefixes are all in T. In the second case we say that the branch is infinite.
4. (Morphisms) A *tree morphism* is any map $\phi : T \to U$, mapping the root of T into the root of U, and any child of any $x \in T$ into some child of $\phi(x) \in U$. ϕ is a *tree isomorphism* if ϕ is bijective.

The root of T is the list $\langle *\rangle$. The children of the root are all lists $\langle *, m_1\rangle \in T$, the children of each $\langle *, m_1\rangle \in T$ are all $\langle *, m_1, m_2\rangle \in T$, and so forth. A branch which is maximal w.r.t. prefix ordering is either a leaf or an infinite branch. We introduce our notion of game.

Definition 2 (*Games*)**.** A game between two players, called $\mathcal{E}$ (Eloise) and $\mathcal{A}$ (Abelard) is any list $G = \langle *, M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}}\rangle$, where:

1. $*$ is a symbol, which we call the start of the game.
2. M is a set, which we call the set of moves of G.
3. T is a tree over M, which we call the set of *positions* of G.
4. $\mathtt{turn}$ is some map : $T \to \{\mathcal{E}, \mathcal{A}\}$.
5. $(W_{\mathcal{E}}, W_{\mathcal{A}})$ is a partition of the infinite branches of T, that is: $W_{\mathcal{E}} \cap W_{\mathcal{A}} = \emptyset$ and $W_{\mathcal{E}} \cup W_{\mathcal{A}} = \{$all infinite branches of $T\}$.

We introduce now some game terminology.

Definition 3 (*Game Terminology*)**.**

1. (Legal Moves) We call $p = \mathtt{turn}(x)$ *the player moving* from position x. If $x, x@\langle m\rangle \in T$, we say that m is a *legal move* of p from position x, a move for short.
2. (Plays and winners) A *play* of G is any finite or infinite branch of T. We say that player $p = \mathcal{E}, \mathcal{A}$ wins the play β if either β is a finite branch of T and $\mathtt{turn}(\beta) \neq p$, or β is an infinite branch and $\beta \in W_p$. The other player is the loser of the play.

We call the elements of T the positions of the game, because from the list of all previous moves we can determine the current status of a play. The root $\langle *\rangle$ of T is called the initial position of the game. The player p moving from position $x \in T$, either $\mathcal{E}$ or $\mathcal{A}$, is by definition $p = \mathtt{turn}(x)$. Player $p_0 = Turn(\langle *\rangle)$ moves from the initial position, selecting some legal move m_1 (some $m_1 \in M$ such that $\langle *, m_1\rangle \in T$). Player $p_1 = \mathtt{turn}(\langle *, m_1\rangle)$ moves next, from position $\langle *, m_1\rangle$, selecting some legal move $m_2 \in M$. The play continues in this way. Players do not always move in an alternating fashion. After k legal moves, the position of the game is $x = \langle *, m_1, \ldots, m_k\rangle \in T$. If x is a leaf, there are no children of x, and therefore no legal moves from x. x is a finite maximal play (without proper extensions). $p = \mathtt{turn}(x)$, the next player moving from x, is the loser of x. We assume that a play may terminate at any moment, even if x is not a leaf, because the player moving next drops out of the play. Therefore each position can also be seen as a complete play. The loser in any finite play x is $p = \mathtt{turn}(x)$, the first player who should move and does not (either because he does not want to, or because he cannot, since x is a leaf). If a player β continues forever, we say that it is won by $\mathcal{E}$ (by $\mathcal{A}$) if $\beta \in W_{\mathcal{E}}$ (if $\beta \in W_{\mathcal{A}}$).

We now define strategies for each player $p = \mathcal{E}, \mathcal{A}$.

Definition 4. Fix any game G as above, a play $\beta = \langle *, m_1, m_2, \ldots\rangle$, a player $p = \mathcal{E}, \mathcal{A}$.

1. A strategy for p, or a p-strategy for short, is any tree σ, such that: (i) $\sigma \subseteq T$; (ii) for all $x \in \sigma$ such that $\mathtt{turn}(x) \neq p$, each $x@\langle m\rangle \in T$ is in σ; (iii) for all $x \in \sigma$ such that $\mathtt{turn}(x) = p$, at most one $x@\langle m\rangle \in T$ is in σ.

2. A p-strategy σ is well-defined on $x \in T$ if either $x \notin \sigma$, or $\mathtt{turn}(x) \neq p$, or x has exactly one child in σ. A p-strategy is well-defined if it is well-defined on all $x \in T$.
3. p follows σ in β if either $\beta \in \sigma$, or β is infinite and all its prefixes are in σ.
4. A strategy is winning for p if p wins all maximal plays in which p follows σ.

If p follows a winning p-strategy, then p wins all plays. If a p-strategy is well-defined, then p at least wins all finite maximal plays β in which p follows σ. *Proof*: Assume by contradiction that $x \in \sigma$, x is a leaf of T, and p loses x. Then $\mathtt{turn}(x) = p$ by definition of winner on x. Therefore, by $x \in \sigma$ and definition of p-strategy, x has one child in $\sigma \subseteq T$, contradicting the assumption that x is a leaf of T.

With each p-strategy σ we can associate a map $\phi : T \to \{*\} \cup M$. If $\phi(x) = m \in M$, we imagine that ϕ is suggesting to p the move m from the position x. If $\phi(x) = *$, we imagine that ϕ is suggesting no move to p from position x.

Definition 5. Let G be as above. Let σ be a p-strategy, and $\phi : T \to \{*\} \cup M$ any map.

1. We say that ϕ is associated with σ if for all $x \in \sigma$ such that $\mathtt{turn}(x) = p$, if $x@\langle\phi(x)\rangle \in T$ then $x@\langle\phi(x)\rangle$ is the unique child of x in σ, while if $x@\langle\phi(x)\rangle \notin T$ then x is a leaf in σ.
2. σ is recursive if some ϕ associated with σ is recursive. σ has recursive degree $\mathcal{O}$ if some ϕ associated with σ has recursive degree $\mathcal{O}$.

In general, there is redundant information in a map $\phi : T \to \{*\} \cup M$ associated with a p-strategy σ. All values of $\phi(x)$ for $x \notin \sigma$, or $\mathtt{turn}(x) \neq p$, are not determined by σ. If $x \in \sigma$ and $\mathtt{turn}(x) = p$, but x is a leaf in σ, then the only information we have concerning $\phi(x)$ is that is not a legal move from x. Only in the case in which $x \in \sigma$, $\mathtt{turn}(x) = p$ and x has some child $y \in \sigma$, we know that y is unique and $\phi(x) = y$. Each map ϕ, instead, is associated with exactly one p-strategy σ.

Lemma 1. *Let G be as above. Let $\phi : T \to \{*\} \cup M$ be a map. Let $\sigma = \{x \in T | \forall y \in T.(y < x \wedge \mathtt{turn}(y) = p) \Rightarrow y@\langle\phi(y)\rangle \leq x\}$ Then σ is a p-strategy, and is the unique p-strategy associated with ϕ.*

Proof. By definition unfolding. □

The previous lemma states that we can define a strategy σ by defining any map $\phi : T \to \{*\} \cup M$, then taking the associated strategy.

The example of game we have in mind are Tarski games for closed positive arithmetical formulas. We consider a language L for first order Arithmetic, having one predicate symbol for each recursive predicate, one function symbol for each recursive map, and the connectives $\vee$, $\exists$, $\wedge$, $\forall$. Let L_0 be the set of closed formulas of L. We define a notion of immediate subformula, and an indexing for immediate subformula, for all $A \in L_0$, as follows. If $A = B_1 \vee \cdots \vee B_n$, $A = B_1 \wedge \cdots \wedge B_n$, the immediate subformulas of A are $B_1, \ldots, B_n$. $1, \ldots, n$ are the indexes of $B_1, \ldots, B_n$. If $A = \exists x.B[x]$, $\forall x.B[x]$, the immediate subformulas of A are all $B[n]$ for $n \in N$. Each n is the index of $B[n]$. The subformula relation is the reflexive and transitive closure of the immediate subformula relation. We define the subformula tree T_A of A as the set of all lists $x = \langle *, n_1, \ldots, n_k\rangle$, such that for some $A_0, \ldots, A_k \in L_0$ we have $A_0 = A$, and for all $i = 1, \ldots, k$, A_i is the immediate subformula of A_{i-1} of index n_i. $A_0, \ldots, A_k \in L_0$ are uniquely determined from x. We say that x is a coding of the subformula A_k of A. The leaves of T_A code the atomic subformulas of A. The root $\langle * \rangle$ of T_A codes A itself.

We now formally define the Tarski game for A. In this game, $\mathcal{E}$ defends the truth of A, and $\mathcal{A}$ defends the falsity of A.

Definition 6. Fix any closed positive arithmetical formula A. Then the Tarski game $\mathcal{G}(A) = \langle *, M, T, W_{\mathcal{E}}, W_{\mathcal{A}}\rangle$ for A is defined as follows:

1. $M = N$ (the set of natural numbers), $* \notin N$.
2. $T =$ the subformula tree T_A of A.
3. If $x \in T_A$ codes B, then we set: $\mathtt{turn}(x) = \mathcal{E}$ if B is atomic false, or a disjunction, or an existential, and: $\mathtt{turn}(x) = \mathcal{A}$ if B is atomic true, or a conjunction, or a universal.
4. $W_{\mathcal{E}} = W_{\mathcal{A}} = \emptyset$ (T has no infinite branches).

The play runs as follows. The initial position $\langle * \rangle$ codes A. Each position x codes some subformula B in A. A legal move from x (if any) consists in selecting some $i \in N$, denoting the i-th subformula B' of B. If $B = C_1 \vee C_2$, or $B = \exists x.C[x]$ then $\mathcal{E}$ moves, selecting some $i = 1, 2$ in the first case, and some $n \in N$ in the second case. Intuitively, $\mathcal{E}$ claims that C_i or $C[n]$ are true. If $B = C_1 \wedge C_2$, or $B = \forall x.C[x]$ then $\mathcal{A}$ moves, selecting some $i = 1, 2$ in the first case, and some $n \in N$ in the second case. Intuitively, $\mathcal{A}$ claims C_i, $C[n]$ are false. If B is atomic then x is a leaf of T_A, and the play ends. If B is false then $\mathcal{E}$ loses, because she should move, and she cannot. If B is true then $\mathcal{A}$ loses, because he should move, and he cannot.

Lemma 2. *$\mathcal{E}$ has a winning strategy on $\mathcal{G}(A)$ if and only if A is true, and $\mathcal{A}$ has a winning strategy on $\mathcal{G}(A)$ if and only if A is false.*

Proof. We may define two canonical maps ϕ, ψ, with ϕ associated to a winning strategy σ for $\mathcal{E}$ when A is true, and with ψ associated to a winning strategy τ for $\mathcal{A}$ if A is false. Assume x codes B. Then we set $\phi(x)$ equal to the index of some true immediate subformula of B, if any, otherwise $\phi(x) = *$. We set $\psi(x)$ equal to the index of some false immediate subformula of B, if any, otherwise $\psi(x) = *$. Notice that, by case analysis, we may check that if $\mathcal{A}$ moves from a true A, then all immediate subformulas of A are true. If $\mathcal{E}$ moves from a true A, then some immediate subformula of A is true. Assume that A is true. By induction over A, we deduce that in all plays following σ, all formulas are true. As a corollary, σ is always defined, and since Tarski games are finite, σ is winning for $\mathcal{E}$ on $\mathcal{G}(A)$. Dually, if A is false, then τ is winning for $\mathcal{A}$ on $\mathcal{G}(A)$. □

The maps ϕ, ψ above and the strategies σ, τ associated with them, are *not* recursive in general. We give an example of a Tarski game with some winning strategy for $\mathcal{E}$, but no recursive winning strategy. Assume that $P(x, y)$ is a recursive predicate, and $\forall y.P(x, y)$ is a non-recursive predicate. For instance, $P(x, y)$ could state that x codes some integer pair $\langle a, b\rangle$, and that the a-th partial recursive map (in some fixed enumeration) applied to b does not converge in y steps or less. Then $\forall y.P(x, y)$ formalizes the Halting problem, a non-recursive statement. Denote with $P^{\perp}$ the complement of P. Let $\mathtt{EM}_1 = \forall x.(\forall y.P(x, y) \vee \exists y.P^{\perp}(x, y))$. $\mathtt{EM}_1$ is equivalent, using intuitionistic reasoning, to Excluded Middle for $\exists y.P^{\perp}(x, y)$. Let G be the Tarski game $\mathcal{G}(\mathtt{EM}_1)$ associated with $\mathtt{EM}_1$. We will now describe G.

The tree T of positions of G is the subformula tree of $\mathtt{EM}_1$. G runs as follows. First, $\mathcal{A}$ moves n. The next position, $\langle *, n\rangle$, codes $\forall y.P(n, y) \vee \exists y.P^{\perp}(n, y)$. Intuitively, this is an instance of $\mathtt{EM}_1$ which $\mathcal{A}$ believes to be false. Then $\mathcal{E}$ moves either $i = 1$ or $i = 2$. The next position, $\langle *, n, i\rangle$, codes either $\forall y.P(n, y)$ or $\exists y.P^{\perp}(n, y)$. Intuitively, this is the component of the disjunction $\mathcal{E}$ believes to be true.

In the first subcase, $\mathcal{A}$ moves $m \in N$. The next position, $\langle *, n, 1, m\rangle$, codes $P(n, m)$, some instance of $\forall y.P(n, y)$ that $\mathcal{A}$ believes to be false. This position is a leaf. The play ends and $\mathcal{E}$ wins if and only if $P(n, m)$ is true.

In the second subcase, $\mathcal{E}$ moves $m \in N$. The next position, $\langle *, n, 2, m\rangle$, codes $P^{\perp}(n, m)$ (notice that there are *two consecutive moves* by $\mathcal{E}$ in this case). This position is a leaf. The play ends, and $\mathcal{E}$ wins if and only if $P^{\perp}(n, m)$ is true.

A map ϕ, associated with any winning strategy for $\mathcal{E}$ on G, must select, from position $\langle *, n\rangle$, the move 1 if $P(n, y)$ is true for all y. If $P(n, m)$ is false (and $P^{\perp}(n, m)$ is true) for some m, then ϕ must select the move 2, then from position $\langle *, n, 2\rangle$ some m such that $P^{\perp}(n, m)$ is true. Therefore, if ϕ associated with a winning strategy for $\mathcal{E}$ on G, then $\phi(\langle *, n\rangle) = 1$ if and only if $\forall y.P(x, y)$ is true. $\forall y.P(x, y)$ is not recursive, therefore ϕ is not recursive. By Definition 5, this means that there is no recursive winning $\mathcal{E}$-strategy. In Section 3, instead, we will introduce a version of $\mathcal{G}(\mathtt{EM}_1)$ with 1-backtracking, in which $\mathcal{E}$ has some recursive winning strategy.

3. Games with 1-backtracking

In this section we define, for all games G (in the sense of Section 2), a game $\mathtt{bck}(G)$ with "1-backtracking". We make precise the informal definition sketched in Section 1. Then we discuss the definition and we introduce one example. More examples can be found in Section 4.

For finite lists x, y, we say that x is a one-step extension of y if $x = y@\langle m\rangle$ for some m. If $\beta = x_0, x_1, x_2, \ldots$ is any sequence of non-empty lists, by $\mathtt{last}(\beta)$ we denote the sequence $m_0 = *, m_1, m_2, \ldots$, where each m_i is the last element of the non-empty list x_i.

Definition 7. Let $G = (*, M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}})$ be a game. Then the 1-backtracking game associated with G is $\mathtt{bck}(G) = (*', M', T', \mathtt{turn}', W'_{\mathcal{E}}, W'_{\mathcal{A}})$, where:

1. $*' = \langle *\rangle$, $M' = T$.
2. T' is the set of finite sequences $\beta = x_0, x_1, \ldots, x_n$ over M' (i.e., over T), such that:
   - $x_0 = \langle *\rangle$;
   - any x_{i+1} is a one-step extension of some x_j such that: (i) $0 \le j \le i$; (ii) $x_j \le x_i$; (iii) $\mathtt{turn}(x_j) = \mathtt{turn}(x_i)$.
3. $\mathtt{turn}'(\langle x_0, \ldots, x_i\rangle) = \mathtt{turn}(x_i)$.
4. (1-limits). If $\beta = x_0, x_1, x_2, \ldots$ is any branch of T', the 1-limit of β is the sequence $\beta^{(1)} = x_{n_0}, x_{n_1}, x_{n_2}, \ldots$, inductively defined as follows.
   - $n_0 = 0$
   - If n_k is any index of $\beta^{(1)}$, and there is a last $j > n_k$ such that x_j is a child of x_{n_k}, then $n_{k+1} = j$. Otherwise, n_k is the last index of $\beta^{(1)}$.
5. $W'_p = \{\beta | (\beta \text{ is an infinite branch of } T) \wedge (p \text{ wins } \mathtt{last}(\beta^{(1)}) \text{ in } G)\}$.

If G is a game in the sense of Section 2, then also $\mathtt{bck}(G)$ is. Therefore we can iterate $\mathtt{bck}(.)$, and define $\mathtt{bck}(\mathtt{bck}(G)), \ldots, \mathtt{bck}^n(G)$. We now discuss the definition of $\mathtt{bck}(G)$.

We first discuss the notion of play with 1-backtracking, that is, of branch of T' (see Definition 7.2). A play with 1-backtracking runs like an ordinary play, except that any player may use the experience he gathered from some wrong move, in order to modify finitely many times some previous move. More in detail, let us assume that the previous moves of the play are $x_0 = \langle m_0\rangle, x_1 = \langle m_0, m_1\rangle, \ldots, x_i = \langle m_0, \ldots, m_i\rangle$. Assume p (either $\mathcal{E}$ or $\mathcal{A}$) is the player moving from position $\langle x_0, \ldots, x_i\rangle$. Then p can play $\langle m_0, \ldots, m_i, m\rangle$, for some legal move m from x_i. However, p can also backtrack to some previous move x_j, provided that: (i) $j < i$; (ii) the list x_j is some prefix of the list x_i (that is, x_j is *an ancestor of* x_i *in the tree of positions of* G); (iii) p moved from x_j. Then p can play $x_{i+1} = \langle m_0, \ldots, m_j, m\rangle$ for some legal move m from x_j.

We present a short example, showing the possibilities and the limitations of 1-backtracking. Assume that $\beta = \langle m_0, m_1, m_2, m_3\rangle$ is some position of a game G. Assume $\mathcal{E}$ moves when the last index is even, and $\mathcal{A}$ when the last index is odd. The last index of β is 3, therefore $\mathcal{A}$ moves next. Rather than moving $\langle m_0, m_1, m_2, m_3, m_4\rangle$, he can backtrack, for instance, to $\langle m_0, m_1\rangle$. Then $\mathcal{A}$ can play $\langle m_0, m_1, m'_2\rangle$. $\mathcal{E}$ moves next. $\mathcal{E}$ may retract a previous move, and backtrack to $\langle m_0\rangle$. However, by Definition 7.2, $\mathcal{E}$ cannot backtrack to $\langle m_0, m_1, m_2\rangle$, since this position is not a prefix of the current position. The first backtracking, by $\mathcal{A}$, removes forever, and for both players, the possibility of backtracking to $\langle m_0, m_1, m_2\rangle$.

We introduce the notion of backtracking-free plays of $\mathtt{bck}(G)$ and we define a copy $G_{\mathtt{bck\text{-}f}}$ of the game G inside G.

Definition 8. Let $\beta = \langle x_0, x_1, x_2, \ldots\rangle$ be a play of $\mathtt{bck}(G)$.

1. x_{i+1} is backtracking-free, or bck.-free short, if it is a one-step extension of x_i. x_{i+1} is a 1-backtracking move if it is a one-step extension of some $x_j < x_i$ with $j < i$. In this case we say that p backtracks to x_j, then moves x_{i+1}.
2. We say that β is bck.-free, if all moves in β are bck.-free. $T'_{\mathtt{bck\text{-}f}}$ is the tree of all finite bck.-free plays of $\mathtt{bck}(G)$. $I_{\mathtt{bck\text{-}f}}$ is the set of all infinite bck.-free plays of $\mathtt{bck}(G)$.
3. We call the game $G_{\mathtt{bck\text{-}f}} = (*', M', T'_{\mathtt{bck\text{-}f}}, \mathtt{turn}', W'_{\mathcal{E}} \cap I_{\mathtt{bck\text{-}f}}, W'_{\mathcal{A}} \cap I_{\mathtt{bck\text{-}f}})$ the bck.-free part of $\mathtt{bck}(G)$.

A play β of $\mathtt{bck}(G)$ is bck.-free if all x_{i+1} in β are one-step extensions of x_i. Thus, β is bck.-free if and only if β is $\langle\langle *\rangle, \langle *, m_1\rangle, \langle *, m_1, m_2\rangle, \langle *, m_1, m_2, m_3\rangle, \ldots\rangle$, for some $m_1, m_2, m_3, \ldots$. Bck.-free positions form a tree $T_{\mathtt{bck\text{-}f}}$ included in the tree of positions of $\mathtt{bck}(G)$, and which is tree-isomorphic to the tree T of positions of G. The isomorphism pair between $T_{\mathtt{bck\text{-}f}}$, T is $\mathtt{last}$, $\mathtt{prefs}$, with $\mathtt{prefs}(x)$ equal to the list $\langle x_0, \ldots, x_n\rangle$ of prefixes of x, in increasing order. If β is bck.-free then $\beta^{(1)} = \beta$ because for any move x_i in β, but the last one, the only move replying to x_i in β is x_{i+1}. Thus, the winning condition for player p in an infinite play β in $G_{\mathtt{bck\text{-}f}}$ is equivalent to $\mathtt{last}(\beta) \in W_p$. We conclude that the backtracking-free game $G_{\mathtt{bck\text{-}f}}$ is, up to an application of the map $\mathtt{last}$, identical to the game G.

We now discuss the notions of 1-limit and winner for an infinite play. Let $\beta = \langle x_0, x_1, x_2, \ldots\rangle$ be any finite or infinite branch of $\mathtt{bck}(G)$, and p the player moving from $x_0 = \langle *\rangle$. First p moves $m \in M$. Later, p can retract his move, by moving $m' \in M$ from x_0. Still later, p can retract his move again, by moving m'' from x_0, and so forth. However, p can only change his move finitely many times, otherwise he loses. If there is no last move from x_0, it is as if there is no move at all from x_0: p loses, because he changed his mind infinitely many times. If we start from the root of the game and we always take the last move from a given position (when it exists), then we remove 1-backtracking from β, defining some bck.-free play $\beta^{(1)}$ which we call the 1-limit of β (see Definition 7.4).

$\beta^{(1)}$ is not computable in general, even when β is computable. Intuitively, $\beta^{(1)}$ includes all definitive choices of both players in β, whenever a definitive choice exists. We have $\beta^{(1)} = \beta$ if and only if β is bck.-free. $\beta^{(1)}$ is some branch of $T_{\mathtt{bck\text{-}f}}$, therefore $\mathtt{last}(\beta^{(1)})$ is some branch of T, i.e., some play of G. In Definition 7.5, the winner of an infinite branch β of $\mathtt{bck}(G)$ is defined as the winner of $\mathtt{last}(\beta^{(1)})$ in G. In general, we cannot decide the winner of an infinite play, not even when the play is recursive. The loser of a finite play β is, by the general definition of game (Section 2, [10]), $p = \mathtt{turn}'(\beta)$, the first player who should move and does not.

We include an example of $\mathtt{bck}(G)$, when G is the Tarski game for $\mathtt{EM}_1$ (see Section 2). The moves of $\mathtt{bck}(G)$ are all positions of $\mathcal{G}(\mathtt{EM}_1)$, that is: $\langle *\rangle$, coding $\mathtt{EM}_1$ itself; all lists $\langle *, n\rangle$, coding $\forall y.P(n, y) \vee \exists y.P^{\perp}(n, y)$; all lists $\langle *, n, 1\rangle$ and $\langle *, n, 2\rangle$, coding: $\forall y.P(n, y)$ and $\exists y.P^{\perp}(n, y)$; all lists $\langle *, n, 1, m\rangle$ and $\langle *, n, 2, m\rangle$, coding: $P(n, m)$ and $P^{\perp}(n, m)$. Following are some examples of legal lists of moves of $\mathtt{bck}(G)$. The sequence of subformulas $\forall y.P(n, y) \vee \exists y.P^{\perp}(n, y)$, $\forall y.P(n, y)$, $P(n, m)$ corresponds to a play both in G and in $\mathtt{bck}(G)$. The list representing the play in G is $\langle *, n, 1, m\rangle$. The list representing the play in $\mathtt{bck}(G)$ is more involved: it is $\langle\langle *\rangle, \langle *, n\rangle, \langle *, n, 1\rangle, \langle *, n, 1, m\rangle\rangle$, a bck.-free play of $\mathtt{bck}(G)$. The difference between $\mathtt{bck}(G)$ and G is that if $P(n, m)$ is false, then in $\mathtt{bck}(G)$ $\mathcal{E}$ does not necessarily lose, but she can play, say, $\langle *, n, 2\rangle$, coding $\exists y.P^{\perp}(n, y)$. This is a legal move because $\mathcal{E}$ backtracks to the prefix $\langle *, n\rangle$ (coding $\forall y.P(n, y) \vee \exists y.P^{\perp}(n, y)$) of the last move $\langle *, n, 1, m\rangle$, and change her previous move $\langle *, n, 1\rangle$ (coding $\forall y.P(n, y)$) to $\langle *, n, 2\rangle$ (coding $\exists y.P^{\perp}(n, y)$). The resulting position in $\mathtt{bck}(G)$ is $\langle\langle *\rangle, \langle *, n\rangle, \langle *, n, 1\rangle, \langle *, n, 1, m\rangle, \langle *, n, 2\rangle\rangle$.

$\mathcal{E}$ has a recursive winning strategy in $\mathtt{bck}(G)$. In order to make the definition of the winning strategy simpler, we first consider the sub-game $\mathtt{bck}_{\mathtt{cf}}(G)$ of $\mathtt{bck}(G)$, in which $\mathcal{A}$ does not backtrack. $\mathtt{bck}_{\mathtt{cf}}(G)$ is defined by taking the subset of positions of $\mathtt{bck}(G)$ in which all moves of $\mathcal{A}$ are backtracking-free. We call $\mathtt{bck}_{\mathtt{cf}}(G)$ the cut-free sub-game of $\mathtt{bck}(G)$. $\mathtt{bck}_{\mathtt{cf}}(G)$ is a game biased in favor of $\mathcal{E}$: she can change finitely many times a move in order to make it better, while $\mathcal{A}$ must choose the right move in the first try. We can imagine $\mathcal{E}$ as a beginner and $\mathcal{A}$ an expert. The bias in favor of $\mathcal{E}$ is a way to allow her to learn how to play better (this trick is indeed used by an expert to teach Chess to a beginner).

Surprisingly, once $\mathcal{E}$ learns how to win in $\mathtt{bck}_{\mathtt{cf}}(G)$, she also wins in the more involved game $\mathtt{bck}(G)$. This result was proved in [4,5]. It has some analogy with the fact that, in Hyland–Ong's game model of PCF, we can restrict ourselves to play "innocently" [9].

Lemma 3. *Every winning strategy τ of $\mathcal{E}$ on $\mathtt{bck}_{\mathtt{cf}}(G)$ may be effectively extended to some winning strategy τ' for $\mathcal{E}$ on $\mathtt{bck}(G)$. τ' is recursive if both τ and the tree of positions of G are recursive.*

For a proof of Lemma 3 we refer to [4]. We informally describe how to extend a strategy for $\mathcal{E}$ from $\mathtt{bck}_{\mathtt{cf}}(G)$ to $\mathtt{bck}(G)$. In any position, $\mathcal{E}$ defines a simplified position by skipping all replies of $\mathcal{A}$ to her moves but the last one. In this way she transforms a position including backtracking moves by $\mathcal{A}$ into one in which there are no backtracking moves by $\mathcal{A}$. Then she can apply her strategy on $\mathtt{bck}_{\mathtt{cf}}(G)$ in order to choose her next move. We claim that in this way a winning strategy on $\mathtt{bck}_{\mathtt{cf}}(G)$ is turned into a winning strategy on $\mathtt{bck}(G)$, and this latter is recursive when the former is.

We now define a recursive winning strategy τ for $\mathcal{E}$ and prove that τ is winning, in the simplified case in which $\mathcal{A}$ does not backtrack. By Lemma 3, we will conclude that we can effectively extend τ to some recursive winning strategy for $\mathcal{E}$.

We present the definition of some recursive map $\psi : T' \to \{*'\} \cup M'$ associated with τ. Three dots $\ldots$ will denote any list of arguments. Assume $\mathcal{A}$ moves $\langle *, n\rangle$, coding $\forall y.P(n, y) \vee \exists y.P^{\perp}(n, y)$. $\mathcal{E}$ first assumes that $\forall y.P(n, y)$ is true and moves $\langle *, n, 1\rangle$: we define $\psi(\langle \ldots, \langle *, n\rangle\rangle) = \langle *, n, 1\rangle$ for all $n \in N$. $\mathcal{A}$ moves $\langle *, n, 1, m\rangle$, for some m: this move codes $P(n, m)$. If $P(n, m)$ is true, then $\mathcal{A}$ loses. If $P(n, m)$ is false, then $\mathcal{E}$ changes her mind (just once!) and she assumes that $\exists y.P^{\perp}(n, y)$ is false. She comes back to $\langle *, n\rangle$, which is *an ancestor of* $\langle *, n, 1, m\rangle$ *in the tree structure of* G. Then $\mathcal{E}$ moves $\langle *, n, 2\rangle$. This is a

1-backtracking move, coding $\exists y.P^{\perp}(n, y)$. We therefore define $\psi(\langle \ldots, \langle *, n, 1, m\rangle\rangle) = \langle *, n, 2\rangle$, for all $n, m \in N$. Eventually, $\mathcal{E}$ moves $\langle *, n, 2, m\rangle$, coding $P^{\perp}(n, m)$, by assumption a true formula, and she wins. We express this last choice by defining $\psi(\langle \ldots, \langle \ldots, m\rangle, \langle *, n, 2\rangle\rangle) = \langle *, n, 2, m\rangle$, for all $n, m \in N$. In any other case, we define $\psi(\beta) = *' = \langle *\rangle$.

We could say that $\mathcal{E}$, during the play, does not know for sure the truth value of $\forall y.P(n, y)$. Rather, she "learns" this truth value by trial-and-error. In Section 4 we include more examples of $\mathtt{bck}(G)$. In Sections 6–8 we prove the results we claimed in Section 1.

4. Some examples of 1-backtracking

In this section we include some examples of games with 1-backtracking. In all examples we consider, $\mathcal{E}$ has some recursive winning strategy on $\mathtt{bck}(G)$, but no recursive winning strategy on G. This supports our thesis that strategies for $\mathtt{bck}(G)$ are "more concrete" than strategies for G. We choose games interpreting well-known mathematical statements.

4.1. Minimum principle for recursive functions

Fix any unary map $g : N \to N$. The minimum value of g is the minimum of the set $\{g(n)|n \in N\}$. $y \in N$ is a minimum point of g if $g(y)$ is a minimum value of g. All maps $g : N \to N$ have a (unique) minimum value and some minimum point. The proof is by contradiction: if it were $\forall y.\exists z.g(y) > g(z)$, then, starting from $y_0 = 0$, we could define an infinite decreasing chain $g(y_0) > g(y_1) > g(y_2) > \cdots$ in N, contradiction. Assume $f : N, N \to N$ is any binary map on N. Define a family $\{f_x\}_{x\in N}$ of unary maps on N by $f_x(y) = f(x, y)$, for all $x, y \in N$. The minimum principle for f states that all f_x's have a minimum point. We may express the minimum principle for f by the (true) formula $M_f = \forall x.\exists y.\forall z.f_x(y) \le f_x(z)$.

Let G_f be the Tarski game for M_f: we call G_f the "Minimum game". The Minimum game G_f runs as follows. First $\mathcal{A}$ moves some $x \in N$. The next position, $\langle *, x\rangle$, codes some instance $\exists y.\forall z.f_x(y) \le f_x(z)$ of M, that $\mathcal{A}$ believes to be false. Then $\mathcal{E}$ moves some $y \in N$. The next position, $\langle *, x, y\rangle$, codes some instance $\forall z.f_x(y) \le f_x(z)$ that $\mathcal{E}$ believes to be true, corresponding to some y that $\mathcal{E}$ believes is a minimum point of f_x. Eventually, $\mathcal{A}$ tests this belief by moving $z \in N$. The next position, $\langle *, x, y, z\rangle$, codes some atomic formula $f_x(y) \le f_x(z)$. If $f_x(y) \le f_x(z)$ is true, then $\mathcal{E}$ wins, otherwise $\mathcal{A}$ wins. Any strategy τ for $\mathcal{E}$ on G_f is associated with some map ψ on the positions of the game. Let $y_x = \psi(\langle *, x\rangle)$, with y_x undefined if $\psi(\langle *, x\rangle) = *'$. The partial map $x \mapsto y_x$ takes any $x \in N$, and returns (when defined) some $y_x \in N$, which $\mathcal{E}$ asserts to be a minimum point of f_x. By definition unfolding, τ is a winning strategy for $\mathcal{E}$ on G_f if and only if y_x is always defined and is always a minimum point of f_x. Since M_f is true, there are winning strategies for $\mathcal{E}$ on G_f. However, $\mathcal{E}$ has no recursive winning strategy in general, unless we allow backtracking.

Lemma 4 (*The Minimum Game*)**.** *Let G_f be the Minimum game for $f : N, N \to N$.*

1. *For some total recursive binary f, there is no recursive winning strategy for $\mathcal{E}$ on G_f.*
2. *For all total recursive binary f, there is some recursive winning strategy τ for $\mathcal{E}$ on $\mathtt{bck}(G_f)$.*

Proof. 1. Let $\mathtt{EM}_1 = \forall x.(\forall y.P(x, y) \vee \exists y.P^{\perp}(x, y))$, for $\forall y.P(x, y)$ some non-recursive predicate, as in Section 2. Define $f_x(y) = 1$ if $P(x, y)$ is true, and $f_x(y) = 0$ if $P(x, y)$ is false. Assume by contradiction that $\mathcal{E}$ has a recursive winning strategy τ for the Minimum Principle for $\{f_x\}_{x\in N}$. Then some associated map ψ is recursive because τ is recursive, and $y_x = \psi(\langle *, x\rangle)$ is a total map because τ is winning. Therefore we can compute one minimum point y_x and the minimum value $f_x(y_x)$ of f_x, given x. f_x has minimum value 1 if and only if $f_x(y) = 1$ for all $y \in N$, and by definition unfolding, if and only if $\forall y.P(x, y)$ is true. Therefore we can compute the truth value of $\forall y.P(x, y)$, which is a contradiction.

2. As we did in Section 2, we define a recursive map ψ associated with τ, then we prove that τ is winning under the assumption that $\mathcal{A}$ does not backtrack. By Lemma 3, we will conclude that τ may be extended to some recursive winning strategy for $\mathcal{E}$ on all plays of $\mathtt{bck}(G_f)$. We present the definition of ψ. Three dots … will denote any list. $\mathcal{A}$ first moves any $\langle *, a\rangle$, that is, he selects some instance $\exists y.\forall z.f_a(y) \le f_a(z)$ he believes to be false. Then $\mathcal{E}$ moves $\langle *, a, b_0\rangle$, with $b_0 = 0$, coding some instance $\forall z.f_a(b_0) \le f_a(z)$ that $\mathcal{E}$ believes to be true. We express this first choice by defining $\psi(\langle \ldots, \langle *, a\rangle\rangle) = \langle *, a, 0\rangle$.

   $\mathcal{A}$ moves $\langle *, a, b_0, b_1\rangle$, selecting some $f_a(b_0) \le f_a(b_1)$ he believes to be false. If $f_a(b_0) \le f_a(b_1)$ is true, $\mathcal{E}$ wins. Otherwise, we know that $f_a(b_0) > f_a(b_1)$. Then $\mathcal{E}$ comes back to $\langle *, a\rangle$ (coding $\exists y.\forall z.f_a(y) \le f_a(z)$), *an ancestor of* $\langle *, a, b_0, b_1\rangle$ *in* G_f), and she moves $\langle *, a, b_1\rangle$, coding $\forall z.f_a(b_1) \le f_a(z)$. $\mathcal{E}$ continues in this way. At each step, either $\mathcal{E}$ wins, or she does not. If $\mathcal{E}$ does not win, then $\mathcal{E}$ backtracks to $\langle *, a\rangle$ and moves $\langle *, a, b_{i+1}\rangle$, if $\langle *, a, b_i, b_{i+1}\rangle$ was the last move by $\mathcal{A}$. We express all these choices by defining $\psi(\langle \ldots, \langle *, a, b, b'\rangle, \rangle) = \langle *, a, b'\rangle$. In any other case, we define $\psi(\beta) = *'$.

   $\mathcal{E}$ defines, as long as she does not win, some sequence $b_0, b_1, b_2, \ldots$ such that $f_a(b_0) > f_a(b_1) > f_a(b_2) > \cdots$. This sequence stops in at most $f_a(b_0) + 1$ steps. The only way this sequence can stop is that $\mathcal{E}$ wins. $\square$

We claim that this recursive winning strategy may be considered as an algorithm "learning by trial-and-error" some minimum point of f_a. At first sight, when the strategy stops, it only provides some b_i such that $f_a(b_i) \le f_a(b_{i+1})$, not some b_i such that $\forall z.f_a(b_i) \le f_a(z)$. However, this strategy is winning not just in $\mathtt{bck}_{\mathtt{cf}}(G_f)$, but also in $\mathtt{bck}(G_f)$. In this latter play, $\mathcal{A}$ can change any number of times his move $\langle *, a, b_i, b_{i+1}\rangle$, choosing in this order $b_{i+1}, b'_{i+1}, b''_{i+1}, b'''_{i+1}, \ldots$. In this case, $\mathcal{E}$ will, *in finite time*, select some i such that $f_a(b_i) \leq f_a(b_{i+1}), f_a(b'_{i+1}), f_a(b''_{i+1}), f_a(b'''_{i+1}), \ldots$, for all $b_{i+1}, b'_{i+1}, b''_{i+1}, \ldots$ selected by $\mathcal{A}$. The main difference with a map selecting a minimum point is that, in the case $\mathcal{A}$ provides infinitely many $b_{i+1}, b'_{i+1}, b''_{i+1}, \ldots$, we never know for sure when $\mathcal{E}$ changes her mind for the last time. We only know that there is, indeed, a last choice of b_i. Suppose, instead, that the "test set" provided by $\mathcal{A}$ for each would-be minimum point of f_a is always finite. In this case we can think of the strategy as an algorithm finding an "empirical" value for the minimum of f_a, depending on some assignment of one finite "test set" to each b_i. This "empirical" minimum point is sometimes different from the real minimum point. However, in many relevant applications of Classical Arithmetic we do not need to compute a real minimum point, but only to compute such an "empirical" minimum point.

4.2. Subsequence principle for recursive functions

We present another example of game G_f, such that $\mathcal{E}$ has a recursive winning strategy for $\texttt{bck}(G_f)$, but no recursive winning strategy for G_f.

G_f is a game corresponding to a combinatorial principle which we call the Subsequence Principle.

Lemma 5 (*Subsequence Principle*). *For all maps $g : N \to N$ there is some increasing sequence $x_1 < x_2 < x_3 < \cdots$ such that $g(x_1) \leq g(x_2) \leq g(x_3) \leq \cdots$.*

Proof. We first Claim that for all $x \in N$ there is some $y \geq x$ such that $g(y) \leq g(z)$ for all $z \geq y$.
Proof of the Claim. We apply the Minimum Principle of Section 4.1 to the map $t \mapsto g(x+t)$.
Now we define $\{x_k\}_{k>0}$ by recursion over k, using the Claim. We set x_1 equal to some $y \geq 0$ such that $g(y) \leq g(z)$ for all $z \geq y$. For all $k > 0$, we set x_{k+1} equal to some $y \geq x_k + 1$ such that $g(y) \leq g(z)$ for all $z \geq y$. By definition of $\{x_k\}_{k>0}$ we have $x_1 < x_2 < x_3 < \cdots$. By definition of $\{x_k\}_{k>0}$ again we deduce $g(x_1) \leq g(y)$ for all $y \geq x_1$, and $g(x_{k+1}) \leq g(y)$ for all $k > 0$ and all $y \geq x_{k+1}$. We conclude that $g(x_1) \leq g(x_2) \leq g(x_3) \leq \cdots$. □

Fix any binary map $f : N, N \to N$ on N. We define a family $\{f_n\}_{n\in N}$ of total unary maps on N by $f_x(y) = f(x, y)$ for all $x, y \in N$. We define the Subsequence principle for f with the statement: $\forall n.\exists x_1 < x_2 < x_3 < \cdots (f_n(x_1) \leq f_n(x_2) \leq f_n(x_3) \leq \cdots)$, having infinitely many quantifiers. We use S to denote this formula. Even if S is an infinite expression, not belonging to our language, we can associate with S a game G_f which we call "the Subsequence game".

The Subsequence game G_f is almost a "solitaire" game: apart from the first move $n \in N$ by $\mathcal{A}$, all moves are by $\mathcal{E}$. The tree T of positions of G_f consists of all finite lists $\langle *, n, x_1, \ldots, x_k\rangle$, such that $x_1 < \cdots < x_k$ and $f_n(x_1) \leq \cdots \leq f_n(x_k)$. The starting position is $\langle *\rangle$. $\mathcal{E}$ wins only all infinite plays. $\mathcal{A}$ wins in all leaves of T (i.e., in all $\langle *, n, x_1, \ldots, x_k\rangle \in T$ such that $f_n(x_k) > f_n(x)$ for all $x > x_k$).

There is a winning strategy for $\mathcal{E}$ on G_f, defining for each first move $n \in N$ of $\mathcal{A}$ an infinite branch of T, corresponding to some infinite increasing sequence $x_1 < x_2 < x_3 < \cdots$ such that $f_n(x_1) \leq f_n(x_2) \leq f_n(x_3) \leq \cdots$. Instead, $\mathcal{E}$ has a recursive winning strategy on G_f only if we allow backtracking.

Lemma 6 (*The Subsequence Game*). *Let $f : N, N \to N$ be a binary map, and G_f be the Subsequence game for f.*

1. *For some total recursive f, there is no recursive winning strategy for $\mathcal{E}$ on G_f.*
2. *For all total recursive f, there is some recursive winning strategy τ for $\mathcal{E}$ on $\texttt{bck}(G_f)$.*

Proof. 1. Assume that $P(x, y)$ is a recursive predicate, and $\forall y.P(x, y)$ is a non-recursive predicate. Define $f_x(y) = 1$ if $P(x, 0) \wedge \cdots \wedge P(x, y)$, and $f_x(y) = 0$ otherwise. Then all f_x are weakly decreasing. If $x_1 < x_2 < x_3 < \cdots$ is any infinite sequence such that $f_x(x_1) \leq f_x(x_2) \leq f_x(x_3) \leq \cdots$, then $f_x(x_1) = f_x(x_2) = f_x(x_3) = \cdots$. If $f_x(x_1) = 1$, then $P(x, y)$ is true for all y, and if $f_x(x_1) = 0$, then $P(x, y)$ is false for some y. Assume there is a recursive map $\psi : M \to T$ associated with a winning strategy for $\mathcal{E}$. Then $x_1 = \psi(\langle *, x\rangle)$ is the first element of the infinite subsequence, and by computing $f_x(x_1)$ we compute the truth value of $\forall y.P(x, y)$, which leads to a contradiction.

2. As usual, we define some associated map ψ. Consider any $\langle *, n, x_1, \ldots, x_k\rangle \in T$, for any $k \in N$. Set $x = 0$ if $k = 0$, and $x = x_k + 1$ otherwise. Then $\mathcal{E}$ moves $\langle *, n, x_1, \ldots, x_i, x\rangle$, for the last $1 \leq i \leq k$ such that $f_n(x_i) \leq f_n(x)$ ($\mathcal{E}$ moves $\langle *, n, x\rangle$ if there is no such i). We define $\psi(\langle \ldots, \langle *, n, x_1, \ldots, x_k\rangle\rangle) = \langle *, n, x_1, \ldots, x_i, x\rangle$. We have $x_1 < \cdots < x_i < x$ by $x_i \leq x_k < x$, and $f_n(x_1) \leq \cdots \leq f_n(x_i) \leq f_n(x)$ by the choice of x. In the case in which $i < k$, by definition we have $f_n(x_k) > f_n(x)$, and the move requires 1-backtracking: $\langle *, n, x_1, \ldots, x_i\rangle$ is a proper ancestor of $\langle *, n, x_1, \ldots, x_k\rangle$ in T. Let β be the infinite list of nodes of T recursively defined by the strategy τ.

   We *Claim*: for all x_i in $\beta^{(1)}$, x_i has at least the child x_{i+1}, and x_i has finitely many children in β.
   Proof of the Claim. If x_{i+1} were the child of x_j for some $j < i$, then x_{i+1} and not x_{j+1} would be the last child of x_j in β, contradicting the definition of $\beta^{(1)}$. Assume that $\mathcal{E}$ plays $\langle *, n, x_1, \ldots, x_{i+1}\rangle$, $\langle *, n, x_1, \ldots, x_i, x'_{i+1}\rangle$, $\langle *, n, x_1, \ldots, x_i, x''_{i+1}\rangle, \ldots$ from $\langle *, x_1, \ldots, x_i\rangle$. All moves $x_{i+1}, x'_{i+1}, x''_{i+1}, \ldots$ but the first one require 1-backtracking, therefore we have $f_n(x_{i+1}) > f_n(x'_{i+1}) > f_n(x''_{i+1}) > \cdots$. Thus, $\mathcal{E}$ can backtrack at most $f_n(x_{i+1})$ times to $\langle *, n, x_1, \ldots, x_i\rangle$.

   By the Claim and by definition of $\beta^{(1)}$, we deduce that $\beta^{(1)}$ is infinite. Thus, $\texttt{last}(\beta^{(1)})$ is infinite. $\mathcal{E}$ wins $\texttt{last}(\beta^{(1)})$, because $\mathcal{E}$ wins all infinite plays in G_f. By definition, $\mathcal{E}$ wins β in $\texttt{bck}(G_f)$. □

The recursive winning strategy defined in the previous Lemma may be considered an algorithm which "learns by trial-and-error" some infinite sequence as required. Indeed, $\mathcal{E}$ eventually produces some last choice for x_1, then some last choice for x_2, and so forth. Notice that this algorithm is *not* an algorithm returning an infinite branch $\langle *, x_1, x_2, x_3, \ldots \rangle$ in T. In general, we cannot decide whether a given choice of $\mathcal{E}$ for x_1 is the last choice, or not. The same holds for the choices of $x_2, x_3, \ldots$.

5. Liveness and 1-backtracking

In this section we introduce Liveness, a property of games, generalizing the fact that players alternate in the game. We need Liveness in the formal statement of Theorem 2, at the end of Section 7.

Fix any infinite play β of $\mathtt{bck}(G)$. Assume that a player p moves only finitely many times in β. In particular, after some point p cannot subsequently retract a move in the case in which he discovers he was wrong. Backtracking is eventually forbidden to p. This is a serious handicap for p. In this case, Theorem 2 may fail, because p can have a winning strategy of degree 1 in G, but no recursive winning strategy in $\mathtt{bck}(G)$. An example. Consider again the Tarski game G for $\mathtt{EM}_1$. Define the following variant G' of G. G' is obtained by replacing each leaf of the tree of G in which $\mathcal{E}$ loses by one infinite branch, consisting of infinitely many dummy moves of $\mathcal{A}$. $\mathcal{E}$ loses all infinite plays of G'. We may easily prove that $\mathcal{E}$ has a winning strategy of degree 1 in G'. This strategy is essentially the same strategy that $\mathcal{E}$ has in G. However, $\mathcal{E}$ has no recursive winning strategy in $\mathtt{bck}(G')$, because $\mathcal{A}$ may prevent her from backtracking just by playing dummy moves forever, and in this case $\mathcal{A}$ wins.

In order to prevent the failure of Theorem 2, we grant some advantage to a player p in this unfavorable case. We add, in the statement of Theorem 2, the "Liveness" condition on G and p:

Definition 9 (*Liveness*). A game G satisfies Liveness for p if: *"for all infinite plays β of G, if p moves only finitely many times in β, then p wins β"*.

We may reformulate Liveness for G and p as follows: *"if the player p performs infinitely many consecutive moves in a play β of G, then p loses in β"*.

If players alternate in G, or if all plays of G are finite (as in Tarski games), then no player performs infinitely many consecutive moves in G. In this case, both players trivially satisfy Liveness. If G is Tarski game for $\mathtt{EM}_1$ (Section 2), then, indeed, both players can perform infinitely many consecutive moves in $\mathtt{bck}(G)$, by backtracking always to the same position of G. However, in such a case they lose, as the Liveness condition requires. The same holds if G is the game of Section 4.1. If G is the game of Section 4.2, then $\mathcal{E}$ trivially satisfies Liveness, because all moves are by $\mathcal{E}$. However, $\mathcal{A}$ trivially does *not* satisfy Liveness in such a G: $\mathcal{A}$ never moves, yet he always loses any infinite play. Liveness is preserved while forming $G_{\mathtt{bck\text{-}f}}$, $\mathtt{bck}(G)$:

Lemma 7. *If p satisfies Liveness in G, then p satisfies Liveness in $G_{\mathtt{bck\text{-}f}}$ and in $\mathtt{bck}(G)$.*

Proof. p satisfies Liveness in $G_{\mathtt{bck\text{-}f}}$ because $G_{\mathtt{bck\text{-}f}}$ is the isomorphic copy of G in $\mathtt{bck}(G)$. Assume p moves finitely many times in an infinite play β of $\mathtt{bck}(G)$. We will prove that p wins $\beta^{(1)}$, hence $\mathtt{last}(\beta^{(1)})$ in G, and β in $\mathtt{bck}(G)$. We reason by cases over $\beta^{(1)}$. Assume $\beta^{(1)}$ is finite, with last position x_{n_i}. Since β is infinite, x_{n_i} has infinitely many children in β. Since p moves finitely many times, the player moving from x_{n_i} is his opponent, q. Thus, q loses $\beta^{(1)}$ because q is the last player to move. Assume, instead, that $\beta^{(1)}$ is infinite. Since $\beta^{(1)}$ is a subsequence of β, then p moves only finitely many times in $\beta^{(1)}$, and $\beta^{(1)}$ is bck.-free. By the Liveness assumption on $G_{\mathtt{bck\text{-}f}}$, p wins $\beta^{(1)}$. $\square$

We now characterize the elements of a 1-limit play $\beta^{(1)}$ (this is an essential ingredient for the results of our paper).

Lemma 8. *Fix a game $G = (*, M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}})$. Let p be a player, and $\beta = \langle x_0, \ldots, x_n, \ldots \rangle$ a play with 1-backtracking. Assume $n_0, n_1, n_2, \ldots$ is the sequence of indexes of $\beta^{(1)}$ (Definition 7.4). Then:*

1. *For all n_i, $x_{(n_i+1)}$ (if it exists) is a child of x_{n_i}.*
2. *Assume that β has a last element x_m. Then $\beta^{(1)}$ has last index m, last element x_m, and $\beta^{(1)}$ is the list of ancestors of x_m.*
3. *For all $n > 0$, x_n is either a child of x_{n-1}, or the brother of x_i, for some $i < n$.*

Proof. 1. By definition of 1-backtracking, x_{n_i+1} is a child of x_{n_j} for some $j \le i$. We cannot have $j < i$, otherwise x_{n_i+1} would be a child of x_{n_j} in β after $x_{n_{j+1}}$, contradicting the choice of $x_{n_{j+1}}$ as the last child of x_{n_j}. Then x_{n_i+1} is a child of x_{n_i}.

2. Let n_i be the last index of $\beta^{(1)}$. We will prove that $n_i = m$. It will follow that $\beta^{(1)}$ is the list of ancestors of x_m. By contradiction, assume that $n_i < m$. By point 1, x_{n_i} has some child x_{n_i+1} after n_i in β, and therefore some last child $x_{n_{i+1}}$ after n_i. Thus, x_{n_i} is not the last element of $\beta^{(1)}$, which is a contradiction.
3. Let $(\langle x_0, \ldots, x_n \rangle)^{(1)} = \langle x_{n_0}, \ldots, x_{n_k} \rangle$. Then $n_k = n$ by the previous point, and therefore $n = n_k \ge n_{k-1}+1$. If $n = n_{k-1}+1$, then x_n is a child of $x_{n-1} = x_{n_{k-1}}$. If $n > n_{k-1} + 1$, then x_n and (by point 1) $x_{n_{k-1}+1}$ are children of $x_{n_{k-1}}$, therefore they are brothers. $\square$

Two more ingredients for the results of our paper are the following: the definition of non-trivial move in $\mathtt{bck}(G)$, and the relationship between a well-defined strategy in G and a non-trivial move in $\mathtt{bck}(G)$.

Definition 10 (*Non-trivial Moves*)**.** Let G be any game, $\beta = \langle x_0, \dots, x_n \rangle$ any position in $\mathtt{bck}(G)$, and x any move from β in $\mathtt{bck}(G)$.

1. x is *non-trivial* if: either x is a child of x_n, or $x \neq x_i$, for the last $i \leq n$ such that x is a brother of x_i.
2. x is *trivial* in the opposite case.

We perform a trivial move if we backtrack to some previous position, then we *repeat* the last move we did from this position. In the definition of non-trivial moves, we implicitly used the fact that for all moves $x_{n+1} = x$ from a position $\langle x_0, \dots, x_n \rangle$ in $\mathtt{bck}(G)$, either x_{n+1} is a child of x_n, or x_{n+1} has some brother x_i with $i \leq n$. A well-defined strategy on G can be used to produce non-trivial moves in $\mathtt{bck}(G)$.

Lemma 9. *Let G be a game, σ a p-strategy over G, with associated map ϕ, and $\beta = \langle x_0, \dots, x_m \rangle$ be a position of $\mathtt{bck}(G)$. Assume σ is well-defined over all $y \leq x_m$, and $\mathtt{turn}(x_m) = p$. Then for some $i \leq m$ we have: $x_i \leq x_m$, $\mathtt{turn}(x_i) = p$, and $x = x_i @ \langle \phi(x_i) \rangle$ is a non-trivial move from β in $\mathtt{bck}(G)$.*

Proof. We reason by cases.

Assume $x_m \in \sigma$. Then, since $\mathtt{turn}(x_m) = p$ and σ is well-defined over x_m, there is a unique child z of x_m in σ. We have $z = x_m @ \langle \phi(x_m) \rangle$, because ϕ is a map associated with σ, and $x_m @ \langle \phi(x_m) \rangle$ is a non-trivial move from β in $\mathtt{bck}(G)$.

Now assume $x_m \notin \sigma$. By Lemma 8.2, $\beta^{(1)} = \langle x_{n_0}, \dots, x_{n_k} \rangle$ is the list of ancestors of $x_m = x_{n_k}$. Take the last $i \leq k$ such that $x_{n_i} \in \sigma$. By the definition of p-strategy, we have that x_{n_0} is the root of σ. Therefore i exists and $i < k$, because $x_m = x_{n_k} \notin \sigma$. Again by definition of p-strategy, if $\mathtt{turn}(x_{n_i}) \neq p$, then from $x_{n_i} \in \sigma$ and the fact that $x_{n_{i+1}}$ is a child of x_{n_i} we deduce $x_{n_{i+1}} \in \sigma$. This contradicts the choice of i. Thus, $\mathtt{turn}(x_{n_i}) = p$. σ is well-defined over x_{n_i} by assumption. We deduce that $x = x_{n_i} @ \langle \phi(x_{n_i}) \rangle$ exists, and it is the unique child of x_{n_i} in σ. Again by the choice of i we have $x_{n_{i+1}} \notin \sigma$, hence $x_{n_i} @ \langle \phi(x_{n_i}) \rangle \neq x_{n_{i+1}}$. $x_{n_{i+1}}$ is the last child of x_{n_i} in β, and therefore is the last brother of $x_{n_i} @ \langle \phi(x_{n_i}) \rangle$ in β. Thus, $x = x_{n_i} @ \langle \phi(x_{n_i}) \rangle$ is a non-trivial move from β in $\mathtt{bck}(G)$. $\square$

6. Removing 1-backtracking from strategies

In this section we prove that any winning strategy of degree $\mathcal{O}$ for $\mathtt{bck}(G)$ may be transformed into some winning strategy of degree $\mathcal{O}'$ for G. This result does not require the Liveness hypothesis on G. In Section 7 we prove the converse. The converse requires the Liveness hypothesis on G.

We recall what is a recursive degree (for more details we refer to [11]). We say that a map f is Turing-reducible to a map g and we write $f \leq g$ if we may compute f recursively in an oracle for the map g. $\leq$ is a preorder. We say that two maps f and g are Turing equivalent, and we write $f \equiv g$, if $f \leq g$ and $g \leq f$. A recursive degree $\mathcal{O}$ is any equivalence class of the relation $\equiv$. We order Turing degree $\mathcal{O}_1$, $\mathcal{O}_2$ by $\mathcal{O}_1 \leq \mathcal{O}_2$ if and only if $f \leq g$ for any $f \in \mathcal{O}_1$, $g \in \mathcal{O}_2$. Semi-decidable problems in $\mathcal{O}$ are statements of the form $\exists y.(f(x, y) = 1)$, for some binary map $f \in \mathcal{O}$. An oracle for $\exists y.(f(x, y) = 1)$ is some unary map g such that $g(x) = 1$ if and only if $\exists y.(f(x, y) = 1)$. The "jump" $\mathcal{O}'$ of $\mathcal{O}$ is the highest recursive degree of an oracle for semi-decidable problems in $\mathcal{O}'$: for instance, $1 = 0'$ is the recursive degree of an oracle for the Halting problem.

Fix any recursive degree $\mathcal{O}$, and any game $G = (M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}})$ such that $M, T, \mathtt{turn} \in \mathcal{O}$. Let p be any player, and q his opponent. Let $\mathcal{O}'$ be the jump of $\mathcal{O}$. Assume τ of degree $\mathcal{O}$ is some winning strategy for p on $\mathtt{bck}(G)$. Our thesis is that there is a winning strategy σ of degree $\mathcal{O}'$ for p on G. G and $G_{\mathtt{bck\text{-}f}}$ (the set of backtracking-free positions of $\mathtt{bck}(G)$) are clearly isomorphic. Therefore it is enough to define a winning strategy $\sigma \in \mathcal{O}'$ for p in $G_{\mathtt{bck\text{-}f}}$.

By definition unfolding, if p wins β in $\mathtt{bck}(G)$, then p wins $\beta^{(1)}$ in $G_{\mathtt{bck\text{-}f}}$. Our idea, therefore, is to define two plays in parallel, $\beta^{(1)}$ in $G_{\mathtt{bck\text{-}f}}$ and β in $\mathtt{bck}(G)$, such that p follows σ in $\beta^{(1)}$, and follows τ in β. p eventually wins β, hence p will win $\beta^{(1)}$. If p moves from $\beta^{(1)}$, and therefore from β, we have to define the move $\phi(\beta^{(1)})$, and a map ϕ associated with σ. Our goal is to obtain some $\gamma > \beta$ in $\mathtt{bck}(G)$, in which p still follows τ, and such that $\gamma^{(1)} = \beta^{(1)} @ \langle x \rangle$. Then we will set $\phi(\beta^{(1)}) = x$.

Let $\beta = \langle x_0, \dots, x_n \rangle$. A problem is that, for all $\gamma > \beta$, the strategy τ could backtrack to some proper prefix x_i of x_n before moving. In other words, for all $\gamma > \beta$ we could have $\gamma^{(1)} = \langle x_0, \dots, x_i, x \rangle$ for some $i < k$. In this case, x is not a child of x_n. In order to avoid this, we will consider a subset of positions $\beta = \langle x_0, \dots, x_n \rangle$ of $\mathtt{bck}(G)$ which we call *stable*, such that τ never backtracks to some proper prefix of x_n, for any $\gamma > \beta$. In other words, we ask that, if p follows τ, then p never changes his mind about some move in β. Being stable is negatively decidable in $\mathcal{O}$, therefore decidable in $\mathcal{O}'$. Whenever p moves from β, we will search (using blind search and a test for stability in $\mathcal{O}'$), for some stable $\gamma > \beta$ such that $\gamma^{(1)} = \beta^{(1)} @ \langle x \rangle$. It is not evident that such a γ exists. If γ exists, we will set $\phi(\beta^{(1)}) = x$. We call such a γ a stable successor of β. As we shall see, if $\mathtt{turn}(\beta) = p$ and β is stable then, indeed, β has some stable successor γ.

In order to define "stable" plays in $\mathtt{bck}(G)$, we will consider plays in which q does not backtrack.

Definition 11. Let $G = (M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}})$ be any game, and p be any player, and q his opponent. Assume τ is any p-strategy.

1. A play is q-cut-free if q never backtracks (i.e., for all indexes $i > 0$ of α, if $\mathtt{turn}(x_{i-1}) = q$ then x_i is a child of x_{i-1}).
2. We denote by $\mathtt{U}(p, \tau)$ the tree of all finite q-cut-free plays $\alpha = \langle x_0, \dots, x_n \rangle$ of $\mathtt{bck}(G)$, on which p follows τ.
3. $\alpha \in \mathtt{U}(p, \tau)$ is p-stable, stable for short, if for all $\beta \in \mathtt{U}(p, \tau)$ such that $\beta > \alpha$, we have $\beta^{(1)} > \alpha^{(1)}$.

4. β is the stable successor of α if α, β are stable, $\beta > \alpha$ and $\beta^{(1)} = \alpha^{(1)}@\langle x\rangle$ for some move x in $\mathtt{bck}(G)$.

For example, $\langle *\rangle$ is stable (*Proof.* Assume that $\beta^{(1)} = \langle *\rangle$ for some β. $\beta, \beta^{(1)}$ have the same last move $*$ by Lemma 8.2. We conclude that $\beta = \langle *\rangle$).

The elements of $\mathtt{U}(p, \tau)$ are finite lists and may be coded by elements of N. Assume that the membership predicates of T, M and $\mathtt{turn}$ are in $\mathcal{O}$. Then the set $\mathtt{U}(p, \tau)$ is in $\mathcal{O}$. Therefore the predicate "α is stable" is the universal quantification of some predicate in $\mathcal{O}$. Thus, the predicate "α is stable" is in $\mathcal{O}'$. We first check some properties of stable plays we need later.

Lemma 10. *Fix a game $G = (*, M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}})$. Let p be a player, q its opponent. Denote by $\alpha = \langle x_0, \ldots, x_i\rangle$, $\beta = \langle y_0, \ldots, y_j\rangle$, and $\gamma = \langle z_0, \ldots, z_k\rangle$ some positions of $\mathtt{bck}(G)$. Assume that τ is a winning strategy for p on $\mathtt{bck}(G)$, that $\mathtt{U}(p, \tau)$ is as in Definition 11, and that α* is stable. *Then:*

1. *If δ is any (finite or infinite) branch in $\mathtt{U}(p, \tau)$, $\delta \geq \alpha$, then $\delta^{(1)} \geq \alpha^{(1)}$, and the indexes of $\alpha^{(1)}$ are a prefix of the indexes of $\delta^{(1)}$.*
2. *For all $\alpha@\langle x\rangle \in \mathtt{U}(p, \tau)$ we have $(\alpha@\langle x\rangle)^{(1)} = \alpha^{(1)}@\langle x\rangle$, and x is a child of x_i.*
3. *β is a stable successor of α if and only if:* (i) *$\beta \in \mathtt{U}(p, \tau)$, $\beta > \alpha$, y_j is a child of x_i, and* (ii) *for no $\gamma \in \mathtt{U}(p, \tau)$ such that $\gamma > \beta$, we have that z_k is a child of x_i.*

Proof. 1. $\delta^{(1)} \geq \alpha^{(1)}$ follows by definition of stability. Let $\alpha^{(1)} = \langle x_{n_0}, \ldots, x_{n_h}\rangle$. In order to prove the relationship among indexes, it is enough to prove, by induction on $l < h$, that $x_{n_{l+1}}$ is the last child of x_{n_l} in all $\delta \geq \alpha$. By contradiction, we assume that the last child in δ is x_n, for some $n > n_{l+1}$. Let $\theta = \langle x_0, \ldots, x_n\rangle \leq \delta$. Then n is not in α, because n_{l+1} is the last child of x_{n_l} in α. Thus, $n > n_h$, and n_h is (by the Lemma 8.2) the last index of α. Then $\alpha < \theta$. By the induction hypothesis on all indexes $m < l$, we have $\theta^{(1)} = \langle x_{n_0}, \ldots, x_{n_l}, x_n\rangle$. Thus $\theta^{(1)}$ has $l + 2$ elements, and $l + 2 \leq h + 1$ because $l < h$. By the fact that α is stable and $\alpha < \theta$, we also deduce that $\alpha^{(1)} < \theta^{(1)}$. Thus, $\theta^{(1)}$ has more than $h + 1$ elements, which is a contradiction.

2. By Lemma 8.2, the last indexes of $\alpha^{(1)}$, $(\alpha@\langle x\rangle)^{(1)}$ are the last indexes $i, i + 1$ of $\alpha, \alpha@\langle x\rangle$. By point 1 above, the indexes of $\alpha^{(1)}$ are a prefix of the indexes of $(\alpha@\langle x\rangle)^{(1)}$. Thus, $(\alpha@\langle x\rangle)^{(1)} = \alpha^{(1)}@\langle x\rangle$. By Lemma 8.1, we conclude that x is a child of x_i.

3. Assume (i), (ii), $\gamma = \langle z_0, \ldots, z_k\rangle \in \mathtt{U}(p, \tau)$, and $\gamma > \beta > \alpha$, in order to prove $\gamma^{(1)} > \beta^{(1)}$ and $\beta^{(1)} = \alpha^{(1)}@\langle y_j\rangle$. By the fact that α, β are stable, we deduce that $\alpha^{(1)} < \beta^{(1)} < \gamma^{(1)}$. By point 1 above, the indexes of $\alpha^{(1)}$ are a prefix of the indexes of $\beta^{(1)}, \gamma^{(1)}$, and are a proper prefix because $\alpha^{(1)} < \beta^{(1)}, \gamma^{(1)}$. Since y_j is a child of x_i, and x_i is the last element of $\alpha^{(1)}$, then y_j is the last child of x_i in β, and $\beta^{(1)} = \alpha^{(1)}@\langle y_j\rangle$. Since the indexes of $\alpha^{(1)}$ are a proper prefix of the indexes of $\gamma^{(1)}$, y_j is also the last child of x_i in γ, therefore $\gamma^{(1)} \geq \alpha^{(1)}@\langle y_j\rangle = \beta^{(1)}$. By Lemma 8.2, the last element of $\gamma^{(1)}$ is z_k. z_k is not equal to y_j because z_k is not a child of x_i. Thus, $\gamma^{(1)} > \beta^{(1)}$.

The reverse is immediate. □

We now prove that if p has a winning strategy in $\mathtt{bck}(G)$, then any stable play in which p moves next has a stable successor. Using this fact we will define a winning strategy for p in G, which is recursive in $\mathcal{O}'$ if the components of G are in $\mathcal{O}$.

Lemma 11. *Fix any game $G = (*, M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}})$. Let p be any player, q its opponent. Denote by $\alpha = \langle x_0, \ldots, x_i\rangle$, $\beta = \langle y_0, \ldots, y_j\rangle$, and $\gamma = \langle z_0, \ldots, z_k\rangle$ some positions of $\mathtt{bck}(G)$.*

Assume that τ is a winning strategy for p on $\mathtt{bck}(G)$, that $\mathtt{U}(p, \tau)$ is as in Definition 11, and that α is stable. Then:

1. *If $\mathtt{turn}(\alpha) \neq p$, and x is a bck.-free move from α, then $\alpha@\langle x\rangle$ is a stable successor of α.*
2. *If $\mathtt{turn}(\alpha) = p$, then some $\beta > \alpha$ is a stable successor of α.*

Proof. 1. Assume $\mathtt{turn}(\alpha) \neq p$, $\alpha@\langle x\rangle \in \mathtt{bck}(G)$, and that x is a bck.-free move. By the fact that σ is a p-strategy and $\alpha \in U \subseteq \sigma$, we deduce $\alpha@\langle x\rangle \in \sigma$, hence $\alpha@\langle x\rangle \in \mathtt{U}(p, \tau)$ by the fact that x is a bck.-free move. By Lemma 10.3, we have to prove if $\beta > \alpha@\langle x\rangle$, then y_j is not a child of x_i. If it were, since $\mathtt{turn}(x_i) \neq p$, then y_j is a move by q, and a child of y_{j-1} (q does not backtrack on β because $\beta \in \mathtt{U}(p, \tau)$). By the uniqueness of the father, $y_{j-1} = x_i$. Let $\gamma = \langle y_0, \ldots, y_{j-1}\rangle$ be obtained by removing the last element of β. Then $\gamma \geq \alpha@\langle x\rangle$, and $\gamma^{(1)}$ is the list of ancestors of x_i, therefore $\gamma^{(1)} = \alpha^{(1)}$, contradicting the fact that α is stable.

2. By contradiction, assume that $\mathtt{turn}(\alpha) = p$, but that for no $\beta > \alpha$ we have that β is stable and $\beta^{(1)} = \alpha^{(1)}@\langle y_j\rangle$. Since $\mathtt{turn}(\alpha) = p$, and p follows τ in α, there is some $\beta = \alpha@\langle x\rangle \in \mathtt{U}(p, \tau)$. By Lemma 10.2, x is a child of x_i. By Lemma 10.2 and contraposition, for all $\beta \in \mathtt{U}(p, \tau)$ such that $\beta > \alpha$ and y_j child of x_i, there is some $\gamma \in \mathtt{U}(p, \tau)$ such that $\gamma > \beta$ and z_k is a child of x_i. We deduce that there is some infinite sequence $\alpha < \beta = \delta_1 < \delta_2 < \delta_3 < \cdots$ in $\mathtt{U}(p, \tau)$, such that the last element of each δ_n is some child of x_i. There is some infinite branch δ in $\mathtt{U}(p, \tau)$ extending α and all δ_n, and x_i has infinitely many children in δ. By the fact that α is stable and Lemma 10.1, we obtain $\alpha^{(1)} \leq \delta^{(1)}$. Since x_i has infinitely many children in δ, then $\delta^{(1)}$ stops at x_i. p loses in δ since $\mathtt{turn}(x_i) = p$. Since δ is a branch of $\mathtt{U}(p, \tau)$, p follows τ in δ and p wins δ. Therefore we obtain a contradiction. □

Assume that the components of G are in $\mathcal{O}$, and fix any map in $\mathcal{O}$ enumerating $\mathtt{U}(p, \tau)$. By Lemma 11.2, if α is stable there is a first stable $\beta > \alpha$ (in the enumeration of $\mathtt{U}(p, \tau)$) such that $\beta^{(1)} = \alpha^{(1)}@\langle y\rangle$ for some y. We can compute β by minimalization in $\mathcal{O}'$, because being stable is a predicate of $\mathcal{O}'$. We are now ready to state and prove the following theorem.

Theorem 1. *Let $\mathcal{O}$ be any recursive degree. Assume $G = (*, M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}})$ is a game, and $M, T, \mathtt{turn} \in \mathcal{O}$. Let p be a player. Then:*

$$p \text{ wins } \mathtt{bck}(G) \text{ with some strategy } \tau \text{ of degree } \mathcal{O} \Rightarrow p \text{ wins } G \text{ with some strategy } \sigma \text{ of degree } \mathcal{O}'.$$

Proof. We can define σ in $G_{\mathtt{bck\text{-}f}}$ instead of in G, because the two games have a recursive isomorphism. Let $\alpha = \langle x_0, \ldots, x_n\rangle$ be a play of $G_{\mathtt{bck\text{-}f}}$ (equivalently, x_{i+1} is a child of x_i in T, for all $0 \le i < n$). By induction over α, we define two maps $\phi, \Phi \in \mathcal{O}'$, such that, if σ is the strategy associated with ϕ, and p follows σ in α, then $\alpha = \Phi(\alpha)^{(1)}$. If $\alpha = \langle x_0\rangle$, we set $\Phi(\alpha) = \alpha$. Assume $\Phi(\alpha)$ is already defined.

- If $\mathtt{turn}(x_n) = p$ and $\Phi(\alpha)$ is stable, then by Lemma 11.2 we may compute in $\mathcal{O}'$ the first stable $\delta > \Phi(\alpha)$ such that $\delta^{(1)} = \Phi(\alpha)^{(1)}@\langle y\rangle$ for some $y \in G$. We set $\phi(\alpha) = y$ and $\Phi(\alpha@\langle x\rangle) = \delta$, for all children x of x_n.
- Otherwise, we set $\phi(\alpha) = \langle *\rangle$ (a dummy value) and $\Phi(\alpha@\langle x\rangle) = \Phi(\alpha)@\langle x\rangle$, for all children x of x_n.

By induction on β and Lemma 11, we can prove that if $\alpha = \langle x_0, \ldots, x_n\rangle$ is any play of $G_{\mathtt{bck\text{-}f}}$ and p follows σ in α, then $\Phi(\alpha)$ is stable and $\alpha = \Phi(\alpha)^{(1)}$. We can simultaneously prove that $\Phi(\beta) < \Phi(\alpha)$ for all $\beta < \alpha$ (that Φ is an increasing map), and that if $\mathtt{turn}(\alpha) = p$, then $\phi(\alpha)$ is a child of x_n. From this latter we deduce that σ is well-defined, and if p follows σ then p wins all finite plays of $G_{\mathtt{bck\text{-}f}}$. From $\Phi(\alpha)$ stable we deduce $\Phi(\alpha) \in \mathtt{U}(p, \tau)$, therefore p follows τ in α.

We have to check that any infinite play $\alpha = \langle x_0, \ldots, x_n, \ldots\rangle$ of G in which p follows σ is won by p. Let $\alpha_n = \langle x_0, \ldots, x_n\rangle$ for all $n \in N$. Then $\{\Phi(\alpha_n)\}_n$ is an infinite sequence of lists increasing by prefix in $\mathtt{U}(p, \tau)$ (because $\{\alpha_n\}_n$ is increasing, and Φ is an increasing map). Let β be the only infinite extension of all $\Phi(\alpha_n)$. Since p follows τ in β and τ is winning, then p wins $\beta^{(1)}$ in G. By assumption, $\alpha_n = \Phi(\alpha_n)^{(1)} \le \beta^{(1)}$(by the fact that $\Phi(\alpha_n)$ is stable, β is a branch of $\mathtt{U}(p, \tau)$, and Lemma 10.1). Thus, $\alpha \le \beta^{(1)}$. From the fact that α is infinite we conclude that $\alpha = \beta^{(1)}$, and that p wins α.

This ends the proof of Theorem 1. □

7. Adding 1-backtracking to strategies

In this section we prove Theorem 2. We assume that we have fixed some game $G = \langle *, M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}}\rangle$, and some recursive degree $\mathcal{O}$. We denote by $\mathcal{O}'$ the jump of $\mathcal{O}$. We claim that any winning strategy σ of degree $\mathcal{O}'$ for G may be effectively transformed into some winning strategy of degree $\mathcal{O}$ for $\mathtt{bck}(G)$ (more precisely, into some winning strategy using only non-trivial moves). This result requires a weak hypothesis over G, namely the Liveness condition. Liveness condition generalizes the condition that the players alternate (see Section 5).

We first include a proof sketch of Theorem 2, in the particular case of a Tarski game for a prenex formula with four quantifiers, having alternating players, and of $\mathcal{O} = 0$. This proof sketch is taken from [8]. Let F be any Σ^0_4-prenex formula $\exists x_1 \forall y_1 \exists x_2 \forall y_2 R(x_1, y_1, x_2, y_2)$, for some recursive predicate R. The Tarski game G for F is a four-move game. Assume there is a winning strategy of recursive degree 1 for G. By unfolding the definition of winning strategy, there are degree 1 total maps $f()$ and $g(y_1)$ such that $R(f(), y_1, g(y_1), y_2)$ is true for all $y_1, y_2 \in N$. We have to define a recursive strategy for the Tarski game with 1-backtracking for F.

We say that a k-ary map ϕ is the integer limit of a $(k+1)$-ary map ψ, for $n \to \infty$, and we write $\phi(x_1, \ldots, x_n) = \lim_{n\to\infty} \psi(n, x_1, \ldots, x_n)$, if for all $a_1, \ldots, a_k \in N$ there is some n such that for all $m \ge n$ we have $\phi(a_1, \ldots, a_k) = \psi(m, a_1, \ldots, a_k)$. By a result of Classical Recursion Theory ([11]), every map of degree 1 is the integer limit of some map of degree 0. Moreover, if f, g are as above, we can effectively find, from the integer codes for f, g, some integer codes for total recursive maps h, k such that $f() = \lim_{t\to\infty} h(t)$ and $g(y_1) = \lim_{t\to\infty} k(t, y_1)$ for all $y_1 \in N$. We define an $\mathcal{E}$-winning strategy using the maps h, k.

$\mathcal{E}$ moves $h(0)$ for $\exists x_1$, and, after $\mathcal{A}$'s move b_1 for $\forall y_1$, she moves $k(0, b_1)$ for $\exists x_2$. If $\mathcal{E}$ wins after $\mathcal{A}$'s move b_2 for $\forall y_2$, she stops. Otherwise, we define a non-trivial move (see Definition 10) for $\mathcal{E}$. If $\mathcal{E}$ loses, then $R(h(0), b_1, k(0, b_1), b_2)$ is false. There is some $t_1 > 0$ such that either $h(t_1) \ne h(0)$, or $k(t_1, b_1) \ne k(0, b_1)$: otherwise we would have $f() = h(0)$ and $g(b_1) = k(0, b_1)$, and therefore $R(h(0), b_1, k(0, b_1), b_2)$ would be true. $\mathcal{E}$ computes the first $t_1 > 0$ such that either $h(t_1) \ne h(0)$, or $k(t_1, b_1) \ne k(0, b_1)$. If $h(t_1) \ne h(0)$, then $\mathcal{E}$ backtracks to $\exists x_1$. This time, she moves $h(t_1)$ for $\exists x_1$: this is a non-trivial move, because $h(t_1) \ne h(0)$. After $\mathcal{A}$'s move $b_1^{(1)}$ for $\forall y_1$, $\mathcal{E}$ plays $k(t_1, b_1^{(1)})$ for $\exists x_2$ (again, a non-trivial move), then $\mathcal{A}$ plays some $b_2^{(1)}$. If $h(t_1) = h(0)$ but $k(t_1, b_1) \ne k(0, b_1)$, then $\mathcal{E}$ backtracks to $\exists x_2$. This time, she plays $k(t_1, b_1)$ for $\exists x_2$: this is a non-trivial move, because $k(t_1, b_1) \ne k(0, b_1)$. Then $\mathcal{A}$ plays some $b_2^{(1)}$. As $\mathcal{E}$'s first move for $\exists x_1$ we keep $h(t_1) = h(0)$. As $\mathcal{A}$'s first move for $\forall y_1$, we keep $b_1^{(1)} \equiv b_1$. The game continues in this manner. As long as $R(b_1^{(i)}, h(t_i), k(t_i, b_1^{(i)}), b_2^{(i)})$ is false, $\mathcal{E}$ defines some (possibly infinite) sequence $t_0 = 0 < t_1 < t_2 < \cdots$, such that $t_{i+1} > t_i$, and either $h(t_{i+1}) \ne h(t_i)$, or $k(t_{i+1}, b_1^{(i)}) \ne k(t_i, b_1^{(i)})$, where $b_1^{(i)}, b_2^{(i)}$ are defined as $\mathcal{A}$'s last moves for $\forall y_1$ and $\forall y_2$.

By contradiction, assume that $\mathcal{E}$ loses. This holds only when the play continues forever. Then $h(t_i)$ eventually converges to $f()$. Assume $h(t_i) = f()$ for all $i \ge i_0$. $\mathcal{E}$ never backtracks to $\exists x_1$ after i_0, because $h(t_i)$ does not change after i_0. Then $\mathcal{A}$'s move $b \equiv b_1^{(i_0)}$ for $\forall y_1$ is kept as value of $b_1^{(i)}$ for all $i \ge i_0$, since $\mathcal{E}$ never backtracks beyond $\exists x_2$. Eventually, $k(t_i, b)$ converges to $g(b)$ and then $R(b_1^{(i)}, h(t_i), k(t_i, b_1^{(i)}), b_2^{(i)}) \equiv R(b, h(t_i), k(t_i, b), b_2^{(i)}) \equiv R(b, f(), g(b), b_2^{(i)})$ is true, which is a contradiction.

Now we consider the general case (any game and any recursive degree), and we provide a formal proof. By assumption on σ, there is some map ϕ associated with σ such that $\phi \in \mathcal{O}'$ (σ has the same role of f, g in the example above). By a result of Classical Recursion Theory [11], for some $\phi_{(.)}(.) \in \mathcal{O}$ and all $x \in T$, $\phi(x)$ is the integer limit of $\phi_t(x)$, for $t \to \infty$. For any $t \in N$, the map $\phi_t(.)$ is associated with some p-strategy σ_t on G (σ_t has the same role of $h(t)$, $k(t, .)$ in the example above). σ_t can be a bad strategy for p: for instance, σ_t could be not well-defined.

Our goal is to define some $\psi \in \mathcal{O}$, associated with a winning p-strategy τ on $\mathtt{bck}(G)$.

We first check that for all $x \in T$, $t \in N$ there is some $a \geq t$ such that σ_a is well-defined at least on all prefixes of x. Another property we need is that the predicate "σ is well-defined on x" is computable w.r.t. the maps ϕ, $\mathtt{turn}$ and the set T.

Lemma 12. *Let $G = \langle *, M, T, W_{\mathcal{E}}, W_{\mathcal{A}}\rangle$ be a game, and σ a strategy on G with the associated map ϕ.*

1. *the predicate "σ well-defined on x" is computable w.r.t. T, $\mathtt{turn}$, and the restriction of ϕ to all $y \leq x$. Indeed, σ is well-defined on $x \in T$ if and only if: "either $\phi(y) \not\leq x$ for some $y < x$ such that $\mathtt{turn}(y) = p$, or $\mathtt{turn}(x) \neq p$, or $x@\langle\phi(x)\rangle \in T$".*
2. *Assume σ is a well-defined p-strategy on G, and for some $\phi_{(.)}(.)$ and all $x \in N$, $\phi(x)$ is the integer limit of $\phi_t(x)$, for $t \to \infty$. Let x be any position of G. Then for some k and all $a \geq k$, σ_a is well-defined on all prefixes of x.*

Proof. 1. By Lemma 1 and the definition of associated map, the condition of point 1 is equivalent to "either $x \notin \sigma$, or $\mathtt{turn}(x) \neq p$ or x has a unique child in σ". This latter is equivalent to: "if $x \in \sigma$ and $\mathtt{turn}(x) = p$, then x has a unique child in σ", which is the unfolding of "σ well-defined on x".

2. For some k and all $a \geq k$ we have $\phi_a(y) = \phi(y)$ for all prefixes y of x. Fix some $a \geq k$, and some prefix y_0 of x. By assumption, σ is well-defined on y_0. By point 1 above, either $\phi(z) \not\leq y_0$ for some $z < y_0$ such that $\mathtt{turn}(z) = p$, or $\mathtt{turn}(y_0) \neq p$, or $y_0@\langle\phi(y_0)\rangle \in T$. By $\phi_a(y) = \phi(y)$ for all y prefixes of x, we deduce the same condition, but with $\phi(.)$ replaced by $\phi_a(.)$. By point 1 again, σ_a is well-defined on y_0. □

We can now define a map ψ associated with a strategy τ on $\mathtt{bck}(G)$ using the family of maps $\{\phi_t(.)\}_t$.

Definition 12 (*The Map ψ*). Let $G = \langle *, M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}}\rangle$ be a game, with $T, \mathtt{turn} \in \mathcal{O}$. Assume σ is a well-defined p-strategy on G, with associated map $\phi \in \mathcal{O}'$. Assume that for some $\phi_{(.)}(.) \in \mathcal{O}$ and all $x \in N$, $\phi(x)$ is the integer limit of $\phi_t(x)$, for $t \to \infty$. Let $\beta = \langle x_0, \ldots, x_t\rangle$ be a play of $\mathtt{bck}(G)$. Take the first $a \geq t$ such that $\phi_a(.)$ is well-defined on all prefixes of x_t. Then we define a map ψ by

$$\psi(\beta) = x_i@\langle\phi_a(x_i)\rangle$$

for the first $i \leq t$ such that $x_i@\langle\phi_a(x_i)\rangle$ is a non-trivial move from β in $\mathtt{bck}(G)$. We define τ as the only strategy associated with ψ.

We have $\psi \in \mathcal{O}$, because $\phi_{(.)}(.), T, \mathtt{turn} \in \mathcal{O}$, and ψ is recursive in $\phi_{(.)}(.), T, \mathtt{turn}$ by Lemma 12.1. By Lemma 12.2, for all t there is some $a \geq t$ such that σ_a is well-defined on all prefixes of x. Therefore, by Lemma 9, there is some $i \leq t$ such that $x_i@\langle\phi_a(x_i)\rangle$ is a non-trivial move from β. Thus, ψ is a total map. The associated strategy τ is well-defined, because $\psi(\beta)$ is always a legal move from β. We claim that, for all plays β of $\mathtt{bck}(G)$ in which p follows τ, the limit of β is (isomorphic to) a play in which p follows σ in G, provided that p moves infinitely many times in β.

Lemma 13. *Let $\sigma, \tau, \phi, \psi, p, G$ as in the last definition. Fix a finite or infinite play $\beta = \langle x_0, x_1, x_2, \ldots\rangle$ of $\mathtt{bck}(G)$ such that $\beta \in \tau$. Take any x_t such that $\mathtt{turn}(x_t) = p$. Then:*

1. *$x_{t+1} = x_i@\langle\phi_a(x_i)\rangle$ for some $i \leq t$ such that $x_i \leq x_t$, and for some $a \geq t$.*
2. *x_t has finitely many children in β.*
3. *Either p moves only finitely many times, or p follows σ in $\beta^{(1)}$.*

Proof. 1. By the definition of τ and $\beta \in \tau$.

2. For some k and all $h \geq k$ we have $\phi(x_t) = \phi_h(x_t)$. In order to prove that x_t has finitely many children in β, it is enough to prove that x_t has at most one child with index greater than t, k. Assume for contradiction that there are two children $x_{m+1}, x_{m'+1}$ of x_t in β, which are *consecutive* in the set of children of x_t in β, and such that $t, k \leq m < m'$. By definition of 1-backtracking we have $\mathtt{turn}(x_m) = \mathtt{turn}(x_t) = p$ and $\mathtt{turn}(x_{m'}) = \mathtt{turn}(x_t) = p$. By point 1 and the uniqueness of the father x_t of $x_{m+1}, x_{m'+1}$, we deduce that $x_{m+1} = x_t@\langle\phi_a(x_t)\rangle$ and $x_{m'+1} = x_t@\langle\phi_{a'}(x_t)\rangle$, for some $a \geq m$ and $a' \geq m'$. Then $a, a' \geq k$, and therefore $x_{m+1} = x_t@\langle\phi_a(x_t)\rangle = x_t@\langle\phi(x_t)\rangle = x_t@\langle\phi_{a'}(x_t)\rangle = x_{m'+1}$ by the choice of k. By the definition of τ, however, we have $x_{m+1} \neq x_{m'+1}$, because $x_{m+1}, x_{m'+1}$ are consecutive in the set of children of x_t in β, and τ only makes non-trivial moves. Therefore τ cannot make a move $x_{m'+1}$ which is equal to its last brother x_{m+1}. This results in a contradiction.

3. Let $\beta^{(1)} = \langle x_{t_0}, x_{t_1}, x_{t_2}, \ldots\rangle$. Assume that p moves infinitely many times (hence β is infinite). We will prove, by induction over i, that if $\mathtt{turn}(x_{t_i}) = p$, then $x_{t_{i+1}}$ exists and it is equal to $x_{t_i}@\langle\phi(x_{t_i})\rangle$. We first prove that it exists. By Lemma 8.1, x_{t_i+1} is a child of x_{t_i}. If we combine this with point 2, x_{t_i} has a finite non-empty set of children of index greater than t_i. Thus, $x_{t_{i+1}}$ exists, and it is the last child x_m of x_{t_i} in β with $m > t_i$. It remains to check that $x_m = x_{t_i}@\langle\phi(x_{t_i})\rangle$. By contradiction, assume that $x_m \neq x_{t_i}@\langle\phi(x_{t_i})\rangle$. For some k, all $h \geq k$, and all $j \leq i$ we have $\phi(x_{t_j}) = \phi_h(x_{t_j})$. By assumption p moves infinitely many times, therefore there is some move x_h of p, with index $h \geq k, m+1$. Then $x_m \neq x_{t_i}@\langle\phi(x_{t_i})\rangle = x_{t_i}@\langle\phi_h(x_{t_i})\rangle$. By the inductive assumption, $x_{t_{j+1}} = x_{t_j}@\langle\phi(x_{t_j})\rangle = x_{t_j}@\langle\phi_h(x_{t_j})\rangle$ for all $j < i$ such that $\mathtt{turn}(x_{t_j}) = p$. If we combine the last two remarks, then, by the definition of τ, we conclude that $\psi(\langle x_0, \ldots, x_h\rangle) = x_{t_i}@\langle\phi_h(x_{t_i})\rangle = x_{t_i}@\langle\phi(x_{t_i})\rangle$. However, $x_{t_i}@\langle\phi(x_{t_i})\rangle$ is a child of x_{t_i} of index greater than m, contradicting the choice of x_m as last child. □

We are now ready to state and prove the second theorem of the paper. Recall that the Liveness condition trivially holds in the case the two players alternate, or in the case the game is a Tarski game.

Theorem 2. *Let $\mathcal{O}$ be a recursive degree. Assume that $G = (*, M, T, \mathtt{turn}, W_{\mathcal{E}}, W_{\mathcal{A}})$ is a game, with $T, \mathtt{turn} \in \mathcal{O}$. Assume that player p satisfies Liveness in G. Then:*

$$p \text{ wins } G \text{ with some strategy } \sigma \text{ of degree } \mathcal{O}' \Rightarrow p \text{ wins } \mathtt{bck}(G) \text{ with some strategy } \tau \text{ of degree } \mathcal{O}.$$

Proof. Take τ as in Definition 12 and Lemma 13. We will prove that p wins all plays β in which he follows τ. If β is finite, then p wins because τ is well-defined. Assume that β is infinite. By assumption, p satisfies the Liveness Condition in G. By Lemma 7, p satisfies the Liveness condition in $\mathtt{bck}(G)$. Therefore if p moves finitely many times in β, then p wins. Assume p moves infinitely many times. Then p follows σ in $\beta^{(1)}$ by Lemma 13.3. p wins $\beta^{(1)}$ because σ is winning, and therefore p wins β. □

8. 1-backtracking plays and Limit Computable Mathematics

In this section we will prove that the games of the form $\mathtt{bck}(G)$ are a sound and complete semantics for $\mathtt{LCM}$. For a compact definition of $\mathtt{LCM}$ we refer to the final page of [8], or to Appendix.

We first have to make precise which is the minimum recursive degree of a winning strategy for $\mathcal{E}$ on $\mathcal{G}(A)$. Let L be the language for Arithmetic, introduced in Section 2. We define the degree of the formula $A \in L$ as the height $n \in N$ of the subformula tree of A ($n = 0$ if A is atomic). For any recursive predicate P, we define $P^{\perp}$ as the complement of P. For any $A \in L$, we define $A^{\perp}$ by switching P and $P^{\perp}$, $\wedge$ and $\vee$, $\forall$ and $\exists$ in A. $A^{\perp}$ is equivalent to $\neg A$, but $A^{\perp} \in L$ while $(\neg A) \notin L$ (we have no negation in L). There is some $A(x) \in L$ denoting some unary predicate of degree n as formula, whose Turing equivalence class is the recursive degree n. We refer to [11] for proof: the actual choice of $A(x)$ does not matter. We define Excluded Middle for formulas of degree n by $\mathtt{EM}_n = \forall x.(A(x) \vee A(x)^{\perp})$. By construction, $\mathtt{EM}_n \in L_0$ and $\mathtt{EM}_n$ is a formula of degree $n + 2$. We now prove a result about the minimum recursive degree of a winning strategy for a formula $A \in L_0$ of degree $n + 2$.

Lemma 14. *Assume that $A \in L_0$ is a true formula of degree $\leq n + 2$. Then:*

1. *$\mathcal{E}$ has some recursive winning strategy of degree n on $\mathcal{G}(A)$.*
2. *If $A = \mathtt{EM}_n$, then A has degree $n + 2$, and the minimum recursive degree for a winning strategy of $\mathcal{E}$ on $\mathcal{G}(A)$ is exactly n.*

Proof. We use the fact that there is a map of recursive degree n deciding all formulas of L_0 of degree $\leq n$.

1. Let us consider the strategy σ_A defined by $\sigma_A(B) =$ the first true immediate subformula C of B in a given ordering, if any exists, as in Lemma 2. $\sigma_A(B)$ is undefined if there is no true immediate subformula of B. We check that σ_A has recursive degree n. If A has degree $n + 1$ and α is any play of $\mathcal{G}(A)$, then the binary map: $A, \alpha \mapsto \sigma_A(\alpha)$ has recursive degree n, because $\mathcal{E}$ in $\mathcal{G}(A)$ has only to decide formulas of degree $\leq n$, and there is an oracle of recursive degree n for this task. Assume that A has degree $n + 2$, and $\mathcal{A}$ moves first. If $\mathcal{A}$ moves from A to some B, then B is true, and $\mathcal{E}$ wins using the strategy σ_B, which may be computed in the argument B. This defines a winning strategy σ for $\mathcal{E}$ of recursive degree n. Assume that A has degree $n + 2$, and $\mathcal{E}$ moves first. Then there is some true immediate subformula of A: let B be the first such formula. $\mathcal{E}$ wins by playing B from A, then playing according to σ_B. This defines a winning strategy for $\mathcal{E}$ of recursive degree n.
2. Assume $\mathtt{EM}_n = \forall x.P(x) \vee P^{\perp}(x)$, for some predicate P of recursive degree n. $\mathtt{EM}_n$ has degree $n + 1$. A winning strategy σ for $\mathcal{E}$ may be used to decide $P(x)$, as follows. $\mathcal{A}$ first plays $P(m) \vee P^{\perp}(m)$. If $\mathcal{E}$ plays $P(m)$, then she has a winning strategy for $P(m)$ and $P(m)$ is true. If $\mathcal{E}$ plays $P^{\perp}(m)$, then she has a winning strategy for $P^{\perp}(m)$ and $P^{\perp}(m)$ is true. We conclude that $P(x)$ is Turing-reducible to σ, therefore σ has recursive degree $\geq n$. By the previous point, the minimum recursive degree for a winning strategy for $\mathcal{E}$ is n. □

Remark that if $A \in L_0$ is true of degree 0, 1, then there are constant (and therefore recursive) winning strategies for $\mathcal{E}$ on $\mathcal{G}(A)$. We may characterize in terms of $\mathtt{bck}^k(G)$ the arithmetical formulas of L_0 having a winning strategy of recursive degree n, and true arithmetical formulas, and, especially, formulas true in the Limit Realization Interpretation of Arithmetic. Corollary 1.5 was an open problem in [4] (p. 5).

Corollary 1. *Let $A \in L$ be any positive arithmetical formula (i.e., without $\Rightarrow, \neg$), and let $\mathcal{G}(A)$ be the Tarski game associated with A. Let $k, n \in N$. Assume the degree of A (the height of the subformula tree) is $n + 2$. Let $\mathtt{EM}_n$ be Excluded Middle for degree n formulas (see above).*

1. *$\mathcal{E}$ has some winning strategy of recursive degree k on $\mathcal{G}(A) \Leftrightarrow \mathcal{E}$ has some winning recursive strategy on $\mathtt{bck}^k(\mathcal{G}(A))$.*
2. *A is true $\Leftrightarrow \mathcal{E}$ wins $\mathcal{G}(A) \Leftrightarrow \mathcal{E}$ wins $\mathtt{bck}^n(\mathcal{G}(A))$ with some recursive strategy.*
3. *If $A = \mathtt{EM}_n$, then $\mathcal{E}$ wins $\mathcal{G}(A)$, has no recursive winning strategy for $\mathtt{bck}^0(\mathcal{G}(A)), \ldots, \mathtt{bck}^{n-1}(\mathcal{G}(A))$, and has some recursive winning strategy for $\mathtt{bck}^n(\mathcal{G}(A))$.*
4. A is true in the Limit Realization Interpretation if and only if $\mathcal{E}$ wins $\mathtt{bck}(\mathcal{G}(A))$ with some recursive strategy.
5. *The set of A such that $\mathcal{E}$ wins $\mathtt{bck}(\mathcal{G}(A))$ with some recursive strategy is closed under all intuitionistic rules, but not under all classical rules.*

Proof. 1. $\mathcal{G}(A)$ has only finite plays, therefore $\mathcal{E}$ trivially satisfies Liveness in $\mathcal{G}(A)$. By Lemma 7, for all $h \in N$, $\mathcal{E}$ satisfies Liveness in $\texttt{bck}^h(\mathcal{G}(A))$. The thesis follows from Theorems 1 and 2 applied k times.

2. Since the formula A has degree $n+2$, then A is true if and only if $\mathcal{E}$ wins $\mathcal{G}(A)$ with some strategy of recursive degree n (Lemma 14.1). The thesis follows from point 1 above, with $k = n$.
3. The minimum recursive degree for a winning strategy for $\mathcal{E}$ on $\mathcal{G}(\texttt{EM}_n)$ is n (Lemma 14.2). The thesis follows from point 1 above, with $k = n$.
4. A is true in Limit Realization Interpretation if and only if $\mathcal{E}$ wins $\mathcal{G}(A)$ with some strategy of recursive degree 1. The thesis follows from point 1 above, with $k = 1$.
5. Closure under intuitionistic rules follows from point 4, and from the closure of Limit Computability Interpretation under all intuitionistic rules. For the second part, consider the classical theorem $\texttt{EM}_2$. By point 3, with $n = 2$, there is no recursive winning strategy for $\mathcal{E}$ on $\texttt{bck}(\mathcal{G}(\texttt{EM}_2))$. □

9. Conclusions and future works

In this paper we introduced 1-backtracking, a sound and complete game semantics for LCM, which is a proper subset of true arithmetical formulas. We proved that adding 1-backtracking allows to reduce of one unit the recursive degree of a winning strategy for $\mathcal{E}$ in a game. Our semantics is inspired by the idea of learning by trial-and-error, and should be useful for studying the constructive content of some (not all) classical theorems.

There are other results connecting 1-backtracking game semantics with some subsets of classical truth. In [3], Berardi and Yamagata showed that, if we drop Exchange rule from Sequent Calculus with ω-rule, we define a semi-formal system $\texttt{PA}_1$, deriving exactly all positive arithmetical formulas (i.e., without $\Rightarrow$ and $\neg$) valid in 1-backtracking game semantics, or equivalently, valid in LCM. Therefore 1-backtracking may be characterized by a substructural logic. There is a Church–Rosser isomorphism between infinitary proofs of a formula A in $\texttt{PA}_1$ on one side, and winning strategy with 1-backtracking for $\mathcal{E}$ for the associated Tarski game $\mathcal{G}(A)$ on the other side. The idea behind this isomorphism is that if we have a sequent $\Gamma = A_1, \ldots, A_n$ and we drop the possibility of permutating formulas, then we may interpret Γ as the history of some play with 1-backtracking.

Another result connects 1-backtracking and the part of the Excluded Middle Schema used in a classical proof. Berardi and Tatsuta [2] proved that the set of all positive arithmetical formulas valid in the 1-backtracking game semantics is exactly the set of positive arithmetical formulas which are consequences of $\texttt{EM}_1$ in Intuitionistic Heyting Arithmetic with ω-rule. This provides, indirectly, another characterization of LCM. LCM is a subclassical arithmetic in which we have Excluded Middle only for formulas of degree 1.

A last result is about the generalization of the notion of 1-backtracking. Berardi and de' Liguoro [1] analyzed the pointer structure of a play with backtracking in Coquand's game semantics [4,5], and introduced a measure α of complexity for the backtracking used in a play. α can be any ordinal: when $\alpha = 1$, we re-obtain 1-backtracking in the sense of this paper. α-backtracking provides a stratification of Coquand's backtracking in increasing order of complexity.

These results suggest that we should be able to use n-backtracking for a low value of n (say, $n = 1, 2$) to provide simpler constructive interpretation for classical results which do not require in their proof the entire schema of Excluded Middle, but only a part of it (say, $\texttt{EM}_1$ or $\texttt{EM}_2$).

Eventually, we ask ourselves if the results proved for 1-backtracking may be generalized to n-backtracking, for any $n \in N$, and even to α-backtracking for α any ordinal. Fix any positive arithmetical formulas A. We conjecture:

1. if A has degree n, then A is classically true if and only if it $\mathcal{E}$ has a winning strategy for $\mathcal{G}(A)$ using n-backtracking.
2. $\mathcal{E}$ has a winning strategy for $\mathcal{G}(A)$ using n-backtracking if and only if A is a consequence of $\texttt{EM}_n$ in Intuitionistic Heyting Arithmetic with ω-rule.

Appendix. A short introduction to LCM realizability

In this section we briefly introduce the theory of Limit Computable Mathematics, or LCM. This short section is taken from [8].

Fix any recursive enumeration of Δ^0_2, the set of partial recursive maps in an oracle for the Halting Problem. Denote with $\{a\}'(b)$ the result (possibly undefined) of the application of the Δ^0_2-map number a of the enumeration to $b \in N$. Let $\langle ., . \rangle$ be the map coding pairs in N, and π_1, π_2 be the inverse maps, therefore $\pi_i(\langle a_1, a_2 \rangle) = a_i$ for all $a_1, a_2 \in N$.

Definition 13 (*Realization Relation of* LCM)**.** Assume A is any arithmetical formula, in a language extended with a binary map $\{.\}'(.)$, denoting application for codes of Δ^0_2-maps, and with two unary maps π_1, π_2 denoting projections. For any fresh variable a, we define a formula $a\,\mathbf{r}\,A$ in the same language. We read $a\,\mathbf{r}\,A$ as "a realizes A". The definition of $a\,\mathbf{r}\,A$ runs by induction on A.

1. $a\,\mathbf{r}\,(s = t) \equiv (s = t)$
2. $a\,\mathbf{r}\,A \wedge B \equiv \pi_1(a)\,\mathbf{r}\,A \wedge \pi_2(a)\,\mathbf{r}\,B$
3. $a\,\mathbf{r}\,A \vee B \equiv (\pi_1(a) = 1 \wedge \pi_2(a)\,\mathbf{r}\,A) \vee (\pi_1(a) = 2 \wedge \pi_2(a)\,\mathbf{r}\,B)$
4. $a\,\mathbf{r}\,A \rightarrow B \equiv \forall x.\, x\,\mathbf{r}\,A \rightarrow \{a\}'(x)\,\mathbf{r}\,B$

5. $a \mathbf{r} \forall x.A \quad \equiv \forall x.\{a\}'(x) \mathbf{r} A$
6. $a \mathbf{r} \exists x.A \quad \equiv \pi_2(a) \mathbf{r} A[x/\pi_1(a)]$.

Remark that, in the definition above, $s = t$ is the equality predicate of two expressions s, t possibly including the binary map $\{.\}'(.)$. We interpret $s = t$ by "s, t are both undefined, or both defined and equal". In this way we may assign a truth value to each formula $n \mathbf{r} A$, for any closed A and any constant $n \in N$. LCM is the set of closed arithmetical formulas A such that $\exists a.a \mathbf{r} A$. The difference with the standard Realization interpretation is that we consider an enumeration of Δ_2^0-partial maps, instead of Δ_1^0-partial maps (of partial recursive maps). The name LCM, or "Limit Computable Mathematics", comes from the fact that each map in Δ_2^0 is the limit of some recursive map. We say that each map in Δ_2^0 is "limit recursive", for short. Realizers in LCM are limit recursive.

If A is $\rightarrow$-free, we can interpret $a \mathbf{r} A$ by: a defines a Δ_2^0-winning strategy for the Tarski game associated with A. If $a \mathbf{r} A$ holds in the original definition of realization, the one with Δ_1^0-maps, then a defines a *recursive* winning strategy for the Tarski game associated with A.

References

[1] Stefano Berardi, Ugo de'Liguoro, Toward the interpretation of non-constructive reasoning as non-monotonic learning, Information and Computation 207 (1) (2009) 63–81.
[2] Stefano Berardi, Makoto Tatsuta, Positive arithmetic without exchange is a subclassical logic, in: Proceedings of APLAS Conference, 2007, pp. 271–285.
[3] Stefano Berardi, Yoriyuki Yamagata, A sequent calculus for limit computable mathematics, Annals of Pure and Applied Logic 153 (1–3) (2008) 111–126.
[4] Th. Coquand, A semantics of evidence for classical arithmetic (preliminary version), in: Gérard Huet, Gordon Plotkin, Claire Jones (Eds.), Proceedings of the Second Workshop on Logical Frameworks, Edinburgh, 1991, http://www.dcs.ed.ac.uk/home/lego/html/papers.html.
[5] Th. Coquand, A semantics of evidence for classical arithmetic, Journal of Symbolic Logic 60 (1995) 325–337.
[6] S. Hayashi, Mathematics based on incremental learning — excluded middle and inductive inference, Theoretical Computer Science 350 (1) (2006) 125–139.
[7] S. Hayashi, M. Nakata, Towards limit computable mathematics, in: Types for Proofs and Programs, in: LNCS, vol. 2277, Springer, 2001, pp. 125–144.
[8] S. Hayashi, Can Proofs be animated by games?, Fundamenta Informaticae 77 (4) (2007) 331–343.
[9] M. Hyland, L. Ong, On full abstraction for PCF, Information and Computation 163 (2) (2000) 285–408.
[10] W. Hodges, Logic and games, in: Edward N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, Winter 2004, 2004, http://plato.stanford.edu/archives/win2004/entries/logic-games/.
[11] P.G. Odifreddi, Classical Recursion Theory: The Theory of Functions and Sets of Natural Numbers, Vol. 1, in: Studies in Logic and the Foundations of Mathematics, vol. 125, Elsevier, 1999.